Efficient Methods for Large-Scale Unconstrained Optimization

Lukšan, Ladislav
2004

Institute of Computer Science
Academy of Sciences of the Czech Republic

Efficient methods for large-scale unconstrained optimization

L. Lukšan, J. Vlček,

Technical report No. V 915

January 2004

Pod Vodárenskou věží 2, 182 07 Prague 8 phone: +420 2 688 42 44, fax: +420 2 858 57 89,
e-mail:luksan@cs.cas.cz, vlcek@cs.cas.cz

Institute of Computer Science
Academy of Sciences of the Czech Republic

Efficient methods for large-scale unconstrained optimization

L. Lukšan, J. Vlček, [1]

Technical report No. V 915

January 2004

Abstract:

This contribution contains a description of efficient methods for large-scale unconstrained optimization. Many of them have been developed recently by the authors. It concerns limited memory methods for general smooth optimization, variable-metric bundle methods for partially separable nonsmooth optimization, hybrid methods for sparse least squares and methods for solving large-scale trust-region subproblems. These methods are compared with other methods by extensive computational experiments.

Keywords:
Unconstrained optimization, large-scale optimization, nonsmooth optimization, limited-memory methods, bundle-type methods, variable metric methods, nonlinear least squares, nonlinear minimax optimization, trust-region subproblems, computational experiments

[1]This work was supported by the Grant Agency of the Czech Academy of Sciences, project code A1030405, and by the Ministry of Education of the Czech Republic, project code MSM 242200002, L. Lukšan is also from Technical University of Liberec, Hlkova 6, 461 17 Liberec

1 Introduction

Modern numerical methods for unconstrained optimization have been studied and developed since the sixties of the last century. Nevertheless, many new problems and approaches have appeared only recently. It especially concerns general large-scale problems, which challenged the development of limited-memory variable metric methods [24], and structured large-scale problems, which stimulated the development of variable metric methods for partially separable problems [11] and hybrid methods for sparse least-square problems [17]. Additional approaches arose in connection with nonsmooth unconstrained optimization. In this case, various bundle-type methods [14], [15], [22] were developed including variable-metric bundle methods [20], [29], which substantially reduce the size of bundles and, therefore, the number of constraints in the quadratic programming subproblems. Variable-metric bundle methods were recently generalized to solve large-scale nonsmooth problems using a limited-memory variable metric approach [12], [13] or a partially-separable variable metric framework [21]. Furthermore, new methods [10], [18] for solving large-scale trust-region subproblems were proposed, which can be used in connection with the Newton method for general sparse unconstrained optimization or with the Gauss-Newton method for sparse nonlinear least squares.

In this contribution, we deal with the local minimization of the objective function $F : \mathcal{R}^n \to R$. In Section 2, the function F is assumed to be twice continuously differentiable and limited-memory methods are reviewed, including the most recent methods proposed in [30] and [31]. Section 3 is devoted to the nonsmooth optimization. After introducing basic principles of the bundle methods and describing variable-metric bundle methods, we focus our attention on methods for large-scale nonsmooth problems. Section 4 contains a description of hybrid methods for nonlinear least squares and Section 5 is devoted to efficient methods for solving large-scale trust-region subproblems. All the methods presented were carefully tested and compared using extensive computational experiments

2 Limited-memory variable metric methods

Limited-memory variable metric methods can be efficiently used for large-scale unconstrained optimization in case the Hessian matrix is not known or is not sparse. These methods are usually realized in the line-search framework so that they generate a sequence of points $x_k \in \mathcal{R}^n$, $k \in \mathcal{N}$, by the simple process

$$x_{k+1} = x_k + t_k d_k, \tag{2.1}$$

where $d_k = -H_k g_k$ is a direction vector, H_k is a positive definite approximation of the inverse Hessian matrix and $t_k > 0$ is a scalar step-size chosen in such a way that

$$F_{k+1} - F_k \leq \varepsilon_1 \, t_k \, d_k^T g_k, \quad d_k^T g_{k+1} \geq \varepsilon_2 \, d_k^T g_k \tag{2.2}$$

(the weak Wolfe conditions), where $F_k = F(x_k)$, $g_k = \nabla F(x_k)$ and $0 < \varepsilon_1 < 1/2$, $\varepsilon_1 < \varepsilon_2 < 1$. Matrices H_k, $k \in N$, are computed either by using a limited (small)

number of variable metric updates applied to the unit matrix or by updating low dimension matrices. First, we shortly describe two known limited-memory variable metric methods. Then we focus our attention on new shifted limited-memory variable metric methods.

2.1 Limited memory BFGS method

The most known and commonly used limited-memory BFGS (L-BFGS) method [24] works with matrices $H_k = H_k^k$, where $H_{k-m}^k = \gamma_k I$ (usually $\gamma_k = b_{k-1}/a_{k-1}$) and

$$H_{j+1}^k = V_j^T H_j^k V_j + \frac{\rho_j}{b_j} s_j s_j^T, \quad V_j = I - \frac{1}{b_j} y_j s_j^T \tag{2.3}$$

for $k - m \le j \le k - 1$. Here $s_j = x_{j+1} - x_j$, $y_j = g_{j+1} - g_j$, $a_j = y_j^T H_j y_j$, $b_j = y_j^T s_j$ and ρ_j are correction parameters. Matrix $H_k = H_k^k$ need not be constructed explicitly since we need only vector $d_k = -H_k^k g_k$, which can be computed using two recurrences (the Strang formula). First, vectors

$$u_j = -\left(\prod_{i=j}^{k-1} V_i\right) g_k,$$

$k - 1 \ge j \ge k - m$, are computed using the backward recurrence

$$\begin{aligned} \sigma_j &= s_j^T u_{j+1}/b_j, \\ u_j &= u_{j+1} - \sigma_j y_j, \end{aligned}$$

where $u_k = -g_k$. Then vectors

$$v_{j+1} = \frac{b_{k-1}}{a_{k-1}} \left(\prod_{i=k-m}^{j} V_i\right)^T u_{k-m} + \sum_{l=k-m}^{j} \frac{\rho_l}{b_l} \left(\prod_{i=l+1}^{j} V_i\right)^T s_l s_l^T u_{l+1},$$

$k - m \le j \le k - 1$, are computed using the forward recurrence

$$v_{j+1} = v_j + (\rho_j \sigma_j - y_j^T v_j/b_j) s_j,$$

where $v_{k-m} = (b_{k-1}/a_{k-1}) u_{k-m}$. Finally we set $d_k = v_k$. Note that $2m$ vectors s_j, y_j, $k - m \le j \le k - 1$ are used and stored.

Matrix $H_k = H_k^k$, obtained by updates (2.3), can be expressed in the compact form using low order matrices [2]. In this case

$$H_k = \gamma_k I - [S_k, \gamma_k Y_k] M_k [S_k, \gamma_k Y_k]^T, \tag{2.4}$$

where $S_k = [s_{k-m}, \ldots, s_{k-1}]$, $Y_k = [y_{k-m}, \ldots, y_{k-1}]$, and

$$M_k = \begin{bmatrix} (R_k^{-1})^T (C_k + \gamma_k Y_k^T Y_k) R_k^{-1} & -(R_k^{-1})^T \\ -R_k^{-1} & 0 \end{bmatrix}, \tag{2.5}$$

where C_k is a diagonal matrix containing the diagonal part of $S_k^T Y_k$, and R_k is an upper triangular matrix containing the upper triangular part of $S_k^T Y_k$. Again $2m$ vectors s_j, y_j, $k-m \le j \le k-1$ are used and stored.

The above way can be used for some other variable metric updates. The symmetric rank-one (SR1) update can be expressed in the form

$$H_k = \gamma_k I + (S_k - \gamma_k Y_k)(R_k + R_k^T - C_k - \gamma_k Y_k^T Y_k)^{-1}(S_k - \gamma_k Y_k)^T. \qquad (2.6)$$

It is necessary to note that update (2.3) with Strang recurrences is more stable than expressions (2.4)–(2.5). On the other hand, compact-form formulas are very important, since they can be easily inverted (using duality) and applied directly to $B_k = H_k^{-1}$, which is necessary in trust-region approach or in constrained optimization.

2.2 Methods based on reduced Hessian matrices

Another limited-memory variable metric method, proposed in [7], is based on updating reduced Hessian matrices. Let B_k, $k \in N$, be approximations of Hessian matrices obtained by the BFGS method (with $B_1 = I$). If $\mathcal{G}_k$ and $\mathcal{D}_k$ are linear subspaces spanned by the columns of matrices $G_k = [g_1, \ldots, g_k]$ and $D_k = [d_1, \ldots, d_k]$, then $\mathcal{D}_k = \mathcal{G}_k$. Moreover, $B_k v \in \mathcal{G}_k$ for $v \in \mathcal{G}_k$ and $B_k w = w$ for $w \in \mathcal{G}_k^{\perp}$. Let Z_k be a matrix whose columns form an orthonormal basis in $\mathcal{G}_k$ and let $Q_k = [Z_k, W_k]$ be a square orthogonal matrix. The above consideration implies that

$$Q_k^T B_k Q_k = \begin{bmatrix} Z_k^T B_k Z_k & 0 \\ 0 & I \end{bmatrix}, \quad Q_k^T g_k = \begin{bmatrix} Z_k^T g_k \\ 0 \end{bmatrix}$$

and the direction vector can be obtained from the reduced system

$$d_k = Z_k \tilde{d}_k, \quad Z_k^T B_k Z_k \tilde{d}_k = -\tilde{g}_k, \quad \tilde{g}_k = Z_k^T g_k. \qquad (2.7)$$

Thus complete information concerning the variable metric update is contained in the reduced Hessian approximation $Z_k^T B_k Z_k$. We usually use the Choleski decomposition $R_k^T R_k = Z_k^T B_k Z_k$ and update the upper triangular matrix R_k. More details can be found in [6].

Consider now a limited-dimension subspace $\mathcal{D}_k$ spanned by the columns of matrix $D_k = [d_{k-m+1}, \ldots, d_k]$. This subspace is changed on every iteration. Let Z_k be a matrix whose columns form an orthonormal basis in $\mathcal{D}_k$. In efficient implementations of limited-memory methods based on reduced Hessians, matrices Z_k and $Z_k^T B_k Z_k$ are not used explicitly. An upper triangular matrix T_k such that $D_k = Z_k T_k$ and the Choleski decomposition $R_k^T R_k = Z_k^T B_k Z_k$ are used instead. At the first iteration, we set

$$D_1 = [g_1], \quad T_1 = [\|g_1\|], \quad R_1 = [1], \quad \tilde{g}_1 = [\|g_1\|].$$

On every iteration, we first solve two equations $R_k^T R_k \tilde{d}_k = -\tilde{g}_k$, $T_k v_k = \tilde{d}_k$ and set $d_k = D_k v_k$. Then the line-search is performed to obtain a new point $x_{k+1} = x_k + t_k d_k$ and matrices D_k, T_k are changed to correspond to the subspace $\mathcal{D}_k$. Therefore, we replace the last column of D_k by d_k and the last column of T_k by $\tilde{d}_k$. Now a representation of

the subspace $\mathcal{D}_{k+1}$ has to be formed. First, we project the new gradient $g_{k+1} = g(x_{k+1})$ into the subspace $\mathcal{D}_k$ by solving the equation $T_k^T r_{k+1} = D_k^T g_{k+1}$. Then we determine the quantity $\rho_{k+1} = \|g_{k+1}\| - \|r_{k+1}\|$, set $D_{k+1} = [D_k, g_{k+1}]$ and

$$T_{k+1} = \begin{bmatrix} T_k & r_{k+1} \\ 0 & \rho_{k+1} \end{bmatrix}, \quad \tilde{g}_{k+1} = \begin{bmatrix} r_{k+1} \\ \rho_{k+1} \end{bmatrix}.$$

Thus we obtain a temporary representation of the reduced Hessian approximation in the form $Z_{k+1}^T B_k Z_{k+1} = R_{k+1}^T R_{+1}$, where

$$R_{k+1} = \begin{bmatrix} R_k & 0 \\ 0 & \sqrt{1/\gamma_{k+1}} \end{bmatrix}, \quad \tilde{g}_{k+1} = \begin{bmatrix} r_{k+1} \\ \rho_{k+1} \end{bmatrix}.$$

This factorization has to be updated to satisfy the quasi-Newton condition $R_{k+1}^T R_{k+1} \tilde{s}_k = \tilde{y}_k$, where

$$\tilde{s}_k = t_k \begin{bmatrix} \tilde{d}_k \\ 0 \end{bmatrix}, \quad \tilde{y}_k = \tilde{g}_{k+1} - \begin{bmatrix} \tilde{g}_k \\ 0 \end{bmatrix},$$

Numerically stable methods described in [9] can be used for this purpose. If the subspace $\mathcal{D}_{k+1}$ has dimension $m+1$, then it has to be reduced before the new iteration is started. Denote the matrices after such reduction by $\bar{D}_{k+1}$, $\bar{T}_{k+1}$, $\bar{R}_{k+1}$. Then $\bar{D}_{k+1}$ is obtained from D_{k+1} by deleting its first column and matrices $\bar{T}_{k+1}$, $\bar{R}_{k+1}$ can be constructed by using elementary Givens rotations (see [7] for more details).

2.3 Shifted variable metric methods

Consider line-search methods of the form (2.1)–(2.2). Limited-memory variable metric methods based on reduced Hessians use low-rank matrices $H_k = Z_k(Z_k^T B_k Z_k)^{-1} Z_k^T = U_k U_k^T$, where U_k has m columns at most. Thus H_k is singular and the case when d_k is almost perpendicular to g_k can occur. For this reason, it is advantageous to set $H_k = \zeta_k I + U_k U_k^T$, where $\zeta_k > 0$ is a parameter, which is carefully selected in every iteration. In this subsection, we assume that the rank of $A_k = U_k U_k^T$ is $\min(k, n)$ (i.e., $m = n$).

Shifted variable metric methods use matrices $H_k = \zeta_k I + A_k$, $k \in \mathcal{N}$, where $\zeta_k > 0$ and A_k is positive semidefinite. Starting from the zero matrix, these methods generate a sequence of positive semidefinite matrices A_k, $k \in \mathcal{N}$, satisfying the (modified) quasi-Newton condition $A_{k+1} y_k = \varrho_k \tilde{s_k}$, where $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$ and $\tilde{s}_k = s_k - \zeta_{k+1} y_k$. Here ϱ_k is a correction parameter and $\zeta_{k+1} > 0$ is a shift parameter. Update

$$A_{k+1} = A_k + \varrho_k \frac{\tilde{s}_k \tilde{s}_k^T}{\tilde{b}_k} - \frac{A_k y_k y_k^T A_k}{\bar{a}_k} + \frac{\eta_k}{\bar{a}_k} \left(\frac{\bar{a}_k}{\tilde{b}_k} \tilde{s}_k - A_k y_k \right) \left(\frac{\bar{a}_k}{\tilde{b}_k} \tilde{s}_k - A_k y_k \right)^T \qquad (2.8)$$

is used, where $\bar{a}_k = y_k^T A_k y_k$ and $\tilde{b}_k = y_k^T \tilde{s}_k$. The shifted BFGS method corresponds to $\eta_k = 1$. The following theorem is proved in [30].

Theorem 1. Let A_k be positive semidefinite and $\eta_k \geq 0$. If $0 < \zeta_{k+1} < y_k^T s_k / y_k^T y_k$, then A_{k+1} is positive semidefinite.

A crucial part of shifted variable metric methods is the determination of the shift parameter. Theorem 1 implies condition

$$\zeta_{k+1} = \mu_k \, b_k/\hat{a}_k, \quad 0 < \mu_k < 1,$$

where $b_k = y_k^T s_k$ and $\hat{a}_k = y_k^T y_k$. If μ_k is too small, then matrix H_k is unsuitable, especially in the first n iterations, when A_k is singular. If μ_k is too large, the stability is usually lost (numerical explosion). Two basic choices were tested. The simplest choice uses constant $\mu_k = \mu$, $0 < \mu < 1/2$, in every iteration. If $\mu \to 1/2$, then the shifted BFGS method becomes unstable. Efficient values lie in the interval $0.20 \leq \mu \leq 0.25$, e.g., $\mu = 0.22$. A more sophisticated choice, derived by using a theoretical investigation of stability and global convergence (see [30]), is given by the formula

$$\mu_k = \sqrt{1 - \bar{a}_k/a_k} \Big/ \left(1 + \sqrt{1 - b_k^2/(\hat{a}_k |s_k|^2)}\right) \tag{2.9}$$

(the numerator assures the global convergence and the denominator assures the stability).

For proving the global convergence, we need the following assumptions.

Assumption 1. The objective function $f : R^n \to R$ is uniformly convex and has bounded second-order derivatives, i.e.

$$0 < \underline{G} \leq \underline{\lambda}(G(x)) \leq \overline{\lambda}(G(x)) \leq \overline{G} < \infty$$

for all $x \in R^n$, where $\underline{\lambda}(G(x))$ and $\overline{\lambda}(G(x))$ are the lowest and the greatest eigenvalues of the Hessian matrix $G(x)$.

Assumption 2. Parameters ϱ_k and μ_k of the shifted VM method are uniformly positive and bounded, in the sense that

$$0 < \underline{\varrho} \leq \varrho_k \leq \overline{\varrho},$$

$$0 < \underline{\mu} \leq \mu_k \leq \overline{\mu} < 1,$$

for every $k \geq 1$.

The following theorem is proved in [31].

Theorem 2. Consider a shifted variable metric method satisfying Assumption 2 with the line-search fulfilling the weak Wolfe conditions. Let the objective function satisfy Assumption 1. Then, if $0 \leq \eta_k \leq 1$ and $\mu_k^2 \leq 1 - \bar{a}_k/a_k$, one has

$$\liminf_{k \to \infty} \|g_k\| = 0.$$

Remark 1. Condition $\mu_k^2 \leq 1 - \hat{a}_k/a_k$ has been used for the choice of the numerator in (2.9). The denominator in (2.9) minimizes the condition number of H_{k+1} in the first iteration.

Shifted variable metric methods were tested by using a set of 92 relatively difficult test problems with 50 and 200 variables implemented in subroutine `TEST28`, which can

be downloaded from `www.cs.cas.cz/~luksan/test.html`. The results are presented in Table 1, where N is the number of variables, MET is the method used (SBFGS - the shifted BFGS method, SDFP - the shifted DFP method, BFGS - the standard BFGS method, DFP - the standard DFP method), NIT is the total number of iterations, NEV is the total number of function and gradient evaluations, NF is the number of failures for a given set (i.e., the number of problems which were not successfully solved) and TIME is the total computational time in seconds.

N	MET	NIT	NEV	NF	TIME
50	SBFGS	11256	12178	-	1.03
	SDFP	46010	48237	8	3.78
	BFGS	14958	16474	1	1.26
	DFP	79486	84215	35	6.66
200	SBFGS	30429	36080	1	25.11
	SDFP	92799	100461	15	74.88
	BFGS	36099	39991	2	27.21
	DFP	146851	158979	32	113.75

Table 1

The results presented in this table imply the following conclusions:

- The shifted VM methods are competitive with the classic VM methods. They are more efficient than standard implementations of the classic VM methods. However, the classic VM methods can be improved by a suitable scaling, which is problematic in the case of shifted VM methods.
- The shifted VM methods are not intended for solving problems, which can be successfully solved by the classic VM methods. However, these methods are ideal as starting methods for the shifted limited-memory VM methods, which are based on the same idea.

2.4 Shifted limited-memory variable metric methods

Shifted limited-memory variable metric methods use recurrences (2.1)–(2.2) with matrix $H_k = \zeta_k I + A_k = \zeta_k I + U_k U_k^T$, where $n \times m$ matrix U_k is updated by formula $U_{k+1} = V_k U_k$ with a low rank matrix V_k chosen in such a way that the (modified) quasi-Newton condition $A_{k+1} y_k = U_{k+1} U_{k+1}^T y_k = \rho_k \tilde{s}_k$ is satisfied. This condition can be replaced by equations

$$U_{k+1}^T y_k = z_k, \quad U_{k+1} z_k = \varrho_k \tilde{s}_k, \quad z_k^T z_k = \varrho_k \tilde{b}_k. \tag{2.10}$$

The following theorem is proved in [31].

Theorem 3. Let T_k be a symmetric positive definite matrix and $z_k \in R^m$. Denote $\mathcal{U}$ the set of $n \times m$ matrices. Then the unique solution to

$$\min\{y_k^T T_k y_k \, \|T_k^{-1/2}(U_{k+1} - U_k)\|_F^2 : \, U_{k+1} \in \mathcal{U}\}$$

s.t. (2.10) is

$$U_{k+1} = U_k - \frac{T_k y_k}{y_k^T T_k y_k} y_k^T U_k + (\varrho_k \tilde{s}_k - U_k z_k + \frac{y_k^T U_k z_k}{y_k^T T_k y_k} T_k y_k) \frac{z_k^T}{z_k^T z_k} \tag{2.11}$$

($T_k y_k$ and z_k are vector parameters defining a class of shifted limited-memory variable metric methods).

Remark 2. Formula (2.11) can be writen in the form

$$U_{k+1} = \frac{\tilde{s}_k z_k^T}{\tilde{b}_k} + \left(I - \frac{T_k y_k y_k^T}{y_k^T T_k y_k} \right) U_k \left(I - \frac{z_k z_k^T}{z_k^T z_k} \right),$$

which implies

$$U_{k+1} U_{k+1}^T = \rho_k \frac{\tilde{s}_k \tilde{s}_k^T}{\tilde{b}_k} + \left(I - \frac{T_k y_k y_k^T}{y_k^T T_k y_k} \right) U_k \left(I - \frac{z_k z_k^T}{z_k^T z_k} \right) U_k^T \left(I - \frac{y_k y_k^T T_k}{y_k^T T_k y_k} \right).$$

Usually $T_k y_k = \tilde{s}_k$. This choice gives the (full) shifted BFGS method if term $z_k z_k^T / z_k^T z_k$ is omitted.

Using suitable values of the vector parameters we obtain particular methods. Assuming that $T_k y_k$ and $\rho_k \tilde{s}_k - U_k z_k$ are linearly dependent and setting

$$z_k = \vartheta_k U_k^T B_k s_k, \quad \vartheta_k = \pm \sqrt{\varrho_k \tilde{b}_k / \bar{c}_k}. \tag{2.12}$$

we obtain rank 1 variationally derived method (VAR1), where

$$U_{k+1} = U_k - \frac{\varrho_k \tilde{s}_k - \vartheta_k A_k B_k s_k}{\varrho_k \tilde{b}_k - \vartheta_k \bar{b}_k} \left(y_k - \vartheta_k B_k s_k \right)^T U_k, \tag{2.13}$$

which gives the best results for the choice $\operatorname{sgn}(\vartheta_k \bar{b}_k) = -1$.

Using z_k given by (2.12) and setting $T_k y_k = \tilde{s}_k$, we obtain rank 2 variationally derived method (VAR2), where

$$U_{k+1} = U_k - \frac{\tilde{s}_k}{\tilde{b}_k} y_k^T U_k + \left(\varrho_k \frac{\tilde{s}_k}{\vartheta_k} - A_k B_k s_k + \frac{\bar{b}_k}{\tilde{b}_k} \tilde{s}_k \right) \frac{s_k^T B_k U_k}{\bar{c}_k}. \tag{2.14}$$

The efficiency of both these methods significantly depends on the value of the correction parameter ϱ_k. Very good results were obtained with choices $\varrho_k = \nu_k$, $\varrho_k = \varepsilon_k$, $\varrho_k = \sqrt{\nu_k \varepsilon_k}$ and $\varrho_k = \zeta_k / (\zeta_k + \zeta_{k+1})$, where $\nu_k = \mu_k / (1 - \mu_k)$, μ_k is a relative shift parameter and $\varepsilon_k = \sqrt{1 - \bar{a}_k / a_k}$ is the damping factor of μ_k.

Using the above formulas, the following theorem assuring the global convergence of VAR1 and VAR2 is a consequence of Theorem 2 (see [31]).

Theorem 4. Consider a shifted variable metric method VAR1 or VAR2 satisfying Assumption 2 and inequality $\mu_k^2 \leq \zeta_k \hat{a}_k / a_k$ together with the line search (2.1)–(2.2). Let the objective function satisfy Assumption 1. Then if

$$\vartheta_k = -\operatorname{sgn} \bar{b}_k \min \left(C, \sqrt{\varrho_k \tilde{b}_k / \bar{c}_k} \right) \quad \text{or} \quad \vartheta_k = \pm \sqrt{\varrho_k \tilde{b}_k / \bar{c}_k}$$

(for VAR1 or VAR2) hold in all iterations ($C > 0$ can be chosen arbitrarily), one has

$$\liminf_{k\to\infty} \|g_k\| = 0.$$

Shifted limited-memory variable metric methods were tested by using a set of 22 test problems with 1000 and 5000 variables implemented in subroutine `TEST14`, which can be downloaded from `www.cs.cas.cz/~luksan/test.html`. Always 10 vectors (or pairs) were stored for $N = 1000$ and 5 vectors (or pairs) were stored for $N = 5000$. The results are presented in Table 2, where N is the number of variables, MET is the method used (VAR1 - the rank 1 variationally derived method, VAR2 - the rank 2 variationally derived method, LBFGSS - the limited-memory BFGS method with Strang recurrences, LBFGSC - the limited-memory BFGS method with compact matrices, LBFGSR - the limited-memory BFGS method with reduced Hessians, CG - the nonlinear conjugate gradient method), NIT is the total number of iterations, NEV is the total number of function and gradient evaluations, NF is the number of failures for a given set (i.e., the number of problems which were not successfully solved) and TIME is the total computational time in seconds.

N	Method	NIT	NEV	NF	TIME
1000	VAR1	19317	19680	-	13.86
	VAR2	18227	18546	-	13.76
	LBFGSS	20427	21456	-	15.17
	LBFGSC	20555	26003	1	16.55
	LBFGSR	22385	33181	-	24.09
	CG	20520	41049	-	17.91
5000	VAR1	94801	97858	-	8:02.1
	VAR2	85662	87483	-	7:22.6
	LBFGSS	108315	111456	2	9:33.8
	LBFGSC	102313	105828	1	10:32.6
	LBFGSR	98046	154931	-	10:41.4
	CG	69805	168471	1	6:45.3

Table 2

The results presented in this table and our other extensive experiments imply the following conclusions:

- Methods VAR1, VAR2 and LBFGSS are very robust. Methods LBFGSC and LBFGSR are more sensitive to round-off errors and their stability decreases as the number of stored vectors increases.

- Methods VAR1 and VAR2 are very efficient, competitive with the LBFGSS method, for our set of test problems. The LBFGSS method can be better than VAR1 and VAR2 for very ill-conditioned problems.

- Method CG is very efficient for extremely large problems, but it frequently terminates before a required precision is achieved.
- Shifted limited-memory VM methods are still under development. Our limited computational experience indicates that they could be improved by using a more suitable choice of parameters.

3 Methods for large-scale nonsmooth optimization

We assume that objective function $F : R^n \to R$ is locally Lipschitz and we are able to compute a (Clarke) subgradient $g \in \partial F(x)$ at any point $x \in R^n$. Since a locally Lipschitz function is differentiable almost everywhere by the Rademacher theorem, then usually $g = \nabla F(x)$. A special feature of nonsmooth problems is the fact that the gradient $\nabla F(x)$ changes discontinuously and is not small in the neighborhood of a local extremum. Thus the standard optimization methods cannot be used efficiently.

3.1 Principles of bundle methods

Values $F(x^k)$, $g(x^k) \in \partial F(x^k)$ at a single point x^k do not suffice for describing the local properties of the nonsmooth objective function. A bundle of values $F^j = F(y^j)$, $g^j \in \partial F(y^j)$ obtained at trial points y^j, $j \in \mathcal{J}_k \subset \{1, \ldots, k\}$, gives much better information. These values serve for the construction of the piecewise linear function

$$F_L^k(x) = \max_{j \in \mathcal{J}_k} \{F^j + (x - y^j)^T g^j\} = \max_{j \in \mathcal{J}_k} \{F(x^k) + (x - x^k)^T g^j - \alpha_j^k\},$$

where $\alpha_j^k = F(x^k) - F_j^k$, $j \in \mathcal{J}_k$, are linearization errors and $F_j^k = F^j + (x^k - x^j)^T g^j$, $j \in \mathcal{J}_k$. In the convex case, this piecewise linear function is majorized by the objective function and, moreover, $\alpha_j^k \geq 0$ for $j \in \mathcal{J}_k$. To guarantee nonnegativity of these numbers in the nonconvex case, the subgradient locality measures

$$\alpha_j^k = \max \left\{ |F(x^k) - F_j^k|, \gamma (s_j^k)^\nu \right\},$$

where $\gamma > 0$, $\nu \geq 1$ and

$$s_j^k = \|x^j - y^j\| + \sum_{i=j}^{k-1} \|x^{i+1} - x^i\|$$

for $j \in \mathcal{J}_k$, are used instead of linearization errors. Since we can only work with limited-size bundles where $|\mathcal{J}_k| \leq m$ ($|\mathcal{J}_k|$ is the cardinality of set $\mathcal{J}_k$), the set $\mathcal{J}_k$ is usually determined in such a way that $\mathcal{J}_k = \{1, \ldots, k\}$ for $k \leq m$, and $\mathcal{J}_{k+1} = \mathcal{J}_k \cup \{k+1\} \backslash \{k+1-m\}$ for $k \geq m$. In this case, one possibility guaranteeing the global convergence of the bundle method is the use of transformed aggregate values F_a^k, g_a^k, s_a^k and

$$\alpha_a^k = \max \left\{ |F(x^k) - F_a^k|, \gamma (s_a^k)^\nu \right\},$$

which accumulate information from the previous iterations. These values represent a linear function which is added to the set of linear functions contained in the bundle.

New aggregate values $\tilde{F}_a^k$, $\tilde{g}_a^k$, $\tilde{s}_a^k$ are obtained by solving the quadratic programming subproblem (see (3.4)) and are transformed to the next iteration by (3.10).

Direction vector $d^k \in R^n$ is usually obtained as a minimum of the piecewise quadratic function

$$F_Q^k(x) = \frac{1}{2}(x - x^k)^T G^k (x - x^k) + \max\{F_L^k(x), F(x^k) + (x - x^k)^T g_a^k - \alpha_a^k\},$$

where $(1/2)(x-x^k)^T G^k (x-x^k)$ is the regularizing term with symmetric positive definite matrix G^k. This term restricts the size of the direction vector (in a similar way as in the trust region methods). This minimization problem is equivalent to the quadratic programming problem: Minimize function

$$\frac{1}{2} d^T G^k d + v \tag{3.1}$$

subject to

$$-\alpha_j^k + d^T g^j \le v, \quad j \in \mathcal{J}_k, \quad -\alpha_a^k + d^T g_a^k \le v \tag{3.2}$$

(v is an extra variable). The solution of the primal QP subproblem can be expressed in the form

$$d^k = -(G^k)^{-1} \tilde{g}_a^k, \quad v^k = -(d^k)^T G^k d^k - \tilde{\alpha}_a^k, \tag{3.3}$$

where

$$\begin{aligned} \tilde{g}_a^k &= \sum_{j \in \mathcal{J}_k} \lambda_j^k g^j + \lambda_a^k g_a^k, \\ (\tilde{\alpha}_a^k, \tilde{F}_a^k, \tilde{s}_a^k) &= \sum_{j \in \mathcal{J}_k} \lambda_j^k (\alpha_j^k, F_j^k, s_j^k) + \lambda_a^k (\alpha_a^k, F_a^k, s_a^k) \end{aligned} \tag{3.4}$$

and where λ_j^k, $j \in \mathcal{J}_k$, λ_a^k, are corresponding Lagrange multipliers. These Lagrange multipliers are also solutions of the dual QP problem: Minimize function

$$\frac{1}{2} \left(\sum_{j \in \mathcal{J}_k} \lambda_j g^j + \lambda_a g_a^k \right)^T (G^k)^{-1} \left(\sum_{j \in \mathcal{J}_k} \lambda_j g^j + \lambda_a g_a^k \right) + \sum_{j \in \mathcal{J}_k} \lambda_j \alpha_j^k + \lambda_a \alpha_a^k \tag{3.5}$$

subject to

$$\lambda_j \ge 0, \quad j \in \mathcal{J}_k, \quad \lambda_a \ge 0, \quad \textstyle\sum_{j \in \mathcal{J}_k} \lambda_j + \lambda_a = 1. \tag{3.6}$$

The minimum value of the dual function is

$$w^k = \frac{1}{2} (\tilde{g}_a^k)^T (G^k)^{-1} \tilde{g}_a^k + \tilde{\alpha}_a^k = -v^k - \frac{1}{2} (\tilde{g}_a^k)^T (G^k)^{-1} \tilde{g}_a^k. \tag{3.7}$$

Using direction vector d^k, we can compute a new approximation of the minimizer of the objective function. It is usually not possible to just set $x^{k+1} = x^k + d^k$. To guarantee the global convergence of the bundle method, we use a line search procedure which generates two points

$$\begin{aligned} x^{k+1} &= x^k + t_L^k d^k, \\ y^{k+1} &= x^k + t_R^k d^k, \end{aligned}$$

where $0 \leq t_L^k \leq t_R^k \leq 1$ are stepsizes, in such a way that exactly one of the two possibilities, the descent step or the zero step, occurs. The descent step implies the conditions

$$t_R^k = t_L^k > 0, \qquad F(x^k + t_L^k d^k) \leq F(x^k) - \varepsilon_L t_L^k w^k, \tag{3.8}$$

while the zero step implies the conditions

$$t_R^k > t_L^k = 0, \qquad (d^k)^T g(x^k + t_R^k d^k) \geq \alpha^{k+1} - \varepsilon_R w^k \tag{3.9}$$

with

$$\alpha^{k+1} = \max\left\{|F(x^k) - F(x^k + t_R^k d^k) + t_R^k (d^k)^T g(x^k + t_R^k d^k)|, \gamma |t_R^k d^k|^\nu\right\}.$$

Here $0 < \varepsilon_L < 1/2$ and $\varepsilon_L < \varepsilon_R < 1$.

After determining x_{k+1}, it is necessary to transform all values to this point. This is realized by the formulas

$$\begin{array}{rcll} F_j^{k+1} & = & F_j^k + (x^{k+1} - x^k)^T g^j, & j \in J_k, \\ F_a^{k+1} & = & \tilde{F}_a^k + (x^{k+1} - x^k)^T \tilde{g}_a^k, & \\ F_{k+1}^{k+1} & = & F^{k+1} + (x^{k+1} - y^{k+1}) g^{k+1}, & \\ g_a^{k+1} & = & \tilde{g}_a^k, & \\ s_j^{k+1} & = & s_j^k + \|x^{k+1} - x^k\|, & j \in J_k, \\ s_a^{k+1} & = & \tilde{s}_a^k + \|x^{k+1} - x^k\|, & \\ s_{k+1}^{k+1} & = & \|x^{k+1} - y^{k+1}\|. & \end{array} \tag{3.10}$$

It remains to specify the way for determining matrices G^k. To ensure the global convergence of a bundle method, we assume for simplicity that matrices G^k are uniformly positive definite and uniformly bounded (their eigenvalues are positive and lie in the compact interval that does not contain zero). Moreover, if the k-th step is a zero step, then we assume that $G^{k+1} - G^k$ is positive semidefinite. These assumptions are relatively strong, but they can be weakened for individual bundle methods. In the most frequently used proximal bundle method, where matrix G^k is a diagonal of the form $G^k = \sigma^k I$, the above assumptions are satisfied if weights σ^k are positive and lie in the compact interval that does not contain zero and $\sigma^{k+1} \geq \sigma^k$ holds in the zero step. Note that the proximal bundle method requires relatively large bundles ($m \sim n$) to be computationally efficient so that the solution of the quadratic programming subproblem (3.1)–(3.2) is time consuming.

It can be proved under mild assumptions (see e.g. [14]) that the number of consecutive zero steps is finite and that every cluster point of the sequence $\{x^k\}$ is a stationary point of the objective function. This follows from the fact that the norms of aggregate subgradients tend to zero implying $0 \in \partial F(x_k)$, if the number of consecutive zero steps is infinite. An infinite sequence of the descent steps can be investigated by the standard way.

3.2 Variable metric methods for nonsmooth problems

Standard bundle methods require relatively large bundles to be computationally efficient. Therefore, we need to solve quadratic programming subproblems with a relatively large number of constraints. At the same time, standard variable metric methods successfully solve many nonsmooth problems. For this reason, it is advantageous to develop special variable metric methods, which combine good properties of both mentioned approaches. Following [29], we apply variable metric updates with current subgradients to matrix $H^k = (G^k)^{-1}$ (used in (3.5)), which allows us to decrease the bundle dimension significantly. At the same time, we use aggregate subgradients after zero steps and a line search described in the previous subsection to guarantee the global convergence.

Variable metric methods described in this subsection use, for the direction determination, the current subgradient after a descent step and the aggregate subgradient after a zero step. The aggregation procedure uses only three subgradients $g^m \in \partial F(x^k)$, $g^{k+1} \in \partial F(y^{k+1})$, $\tilde{g}^k$ and three subgradient locality measures $\alpha_m = 0$, $\alpha_{k+1} \geq 0$, $\tilde{\alpha}_k \geq 0$ (m is the index of the last descent step and the tilde denotes aggregate quantities). The quadratic programming subproblem (3.5)-(3.6) reduces to the minimization of the function

$$\varphi(\lambda_1, \lambda_2, \lambda_3) = \frac{1}{2} \left\| (H^k)^{1/2} (\lambda_1 g^m + \lambda_2 g^{k+1} + \lambda_3 \tilde{g}^k) \right\|^2 + \lambda_2 \alpha^{k+1} + \lambda_3 \tilde{\alpha}^k, \tag{3.11}$$

where $\lambda_i \geq 0$, $i \in \{1, 2, 3\}$ and $\lambda_1 + \lambda_2 + \lambda_3 = 1$. The optimal values $\lambda_i^k \geq 0$, $i \in \{1, 2, 3\}$ can be computed in a simple way. The new aggregate subgradient and the new aggregate subgradient locality measure are computed from the formulas

$$\tilde{g}^{k+1} = \lambda_1^k g^m + \lambda_2^k g^{k+1} + \lambda_3^k \tilde{g}^k, \quad \tilde{\alpha}^{k+1} = \lambda_2^k \alpha^{k+1} + \lambda_3^k \tilde{\alpha}^k. \tag{3.12}$$

In the first iteration or after a descent step, we set $\tilde{g}^k = g^k$, $\tilde{\alpha}^k = 0$ and $m = k$. The direction vector is determined by formula $d^k = -H^k \tilde{g}^k$. At the same time, we set $w^k = (1/2)(\tilde{g}^k)^T H^k \tilde{g}^k + \tilde{\alpha}^k$. If w^k is sufficiently small, then an approximate solution is found.

Positive semidefiniteness of $H^k - H^{k+1}$ (which is equivalent to positive semidefiniteness of $G^{k+1} - G^k$) after a zero step is usually guaranteed by the SR1 update. Therefore, we use the BFGS update after a descent step and the SR1 update after a zero step. The BFGS update

$$H^{k+1} = H^k + \left(t_L^k + \frac{(u^k)^T H^k u^k}{(u^k)^T d^k} \right) \frac{d^k (d^k)^T}{(u^k)^T d^k} - \frac{H^k u^k (d^k)^T + d^k (u^k)^T H^k}{(u^k)^T d^k},$$

where $u^k = g^{k+1} - g^m$, is used only if $(u^k)^T d^k > 0$. Otherwise we set $H^{k+1} = H^k$. The SR1 update

$$H^{k+1} = H^k - v^k (v^k)^T / (u^k)^T v^k,$$

where $v^k = H^k u^k - t_R^k d^k$, is used only if $(v^k)^T \tilde{g}^k < 0$ (which implies $(u^k)^T v^k > 0$). Otherwise we set $H^{k+1} = H^k$.

Detailed descriptions of variable metric methods for nonsmooth functions can be found in [20] and [29]. The following result is proved in [29].

Theorem 5. Assume that function $F : R^n \to R$ is locally Lipschitz and the level set $\{x \in R^n : \ F(x) \leq F(x_1)\}$ is bounded. Then every cluster point of sequence $\{x_k\}$ generated by the nonsmooth variable metric method is stationary for F.

Two methods for nonsmooth optimization (PBM - the proximal bundle method, NVM - the nonsmooth variable metric method) were tested by using a set of 25 test problems with 2–50 variables implemented in subroutine `TEST19`, which can be downloaded from `www.cs.cas.cz/~luksan/test.html`. The results are presented in Table 3, where P is the number of the problem, NIT is the total number of iterations, NEV is the total number of function and subgradient evaluations and F is the reached function value. The last row contains the summary values and the total computational time (in seconds).

Table 3 demonstrates the high efficiency of the nonsmooth variable metric method. It is competitive with the proximal bundle method measured by the number of iterations, even if it uses bundles of dimension at most 2. Moreover, it is more efficient than the proximal bundle method measured by the computational time, since it does not use the time consuming quadratic programming subproblem (with $m \sim n$ constraints).

	PBM			NVM		
P	NIT	NEV	F	NIT	NEV	F
1	42	45	.38117064D-06	34	34	.27598807D-10
2	18	20	.46154993D-08	15	16	.94894120D-10
3	31	33	1.9522245	17	17	1.9522247
4	14	16	2.0000000	17	17	2.0000000
5	17	19	-3.0000000	20	20	-2.9999996
6	13	15	7.2000014	19	19	7.2000000
7	11	12	-1.4142135	10	10	-1.4142133
8	66	68	-.99999940	55	59	-.99999247
9	13	15	-1.0000000	37	37	-.99999979
10	43	46	-7.9999999	14	14	-7.9999998
11	43	45	-43.999999	38	38	-43.999999
12	27	29	22.600162	40	40	22.600162
13	60	62	-32.348678	52	53	-32.348678
14	154	155	-2.9196975	32	32	-2.9197003
15	92	93	.55981566	81	83	.55981553
16	74	75	-.84140828	89	89	-.84140570
17	160	162	9.7857723	241	241	9.7858732
18	128	143	16.703861	88	89	16.703838
19	150	151	.16712381D-06	123	123	.14683215D-05
20	39	40	.12440972D-12	23	23	.00000000
21	245	251	-638530.48	357	359	-638564.91
22	52	53	.11665945D-11	358	360	.41534959D-05
23	19	20	.51313988D-08	65	66	.32729678D-05
24	27	28	.23412735D-07	67	67	.94570857D-06
25	428	450	32.349182	313	315	32.349159
Σ	1966	2046	TIME = 1.48	2205	2221	TIME = 0.93

Table 3

3.3 Variable metric methods for large-scale nonsmooth problems

Proximal bundle methods are not suitable for solving large-scale nonsmooth problems, since they lead to large-scale quadratic programming subproblems, where constraint Jacobian matrices are usually dense. Nonsmooth variable metric methods described in the previous subsection are also unsuitable, since they use dense variable metric updates. Fortunately, these updates can be replaced by updates based on a limited-memory approach or by updates which utilize sparsity. All other algorithmic details can remain unchanged.

A limited-memory approach is investigated in [12]. The resulting method utilizes matrix (2.4)–(2.5) after a descent step and matrix (2.6) after a zero step. Nevertheless, the updating strategy is not quite simple, since the condition requiring positive semidefiniteness of $H^k - H^{k+1}$ after a zero step considerably complicates a logical structure of the algorithm. Algorithmic details of this method together with encouraging computational results are given in [12]. Global convergence of this method is proved in [13].

We have tested another simple strategy based on a shifted limited-memory variable metric update. In this case, update VAR2 (see (2.14)) is applied after every descent step. It is also used after a zero step if $(\tilde{g}^k)^T H^{k+1} \tilde{g}^k \leq (\tilde{g}^k)^T H^k \tilde{g}^k$. In the opposite case, matrix H^k is kept unchanged.

An efficient method based on partitioned variable metric updates is proposed in [21]. This method has been developed for minimizing partially separable functions of the form

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

where $f_i(x)$, $1 \leq i \leq m$ (m is usually large), are nonsmooth functions depending on a small number of variables (n_i, say). A typical example is

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

(sum of absolute values). If $n_i \ll n$ for $1 \leq i \leq m$, subgradients g_i, generalized Hessian matrices G_i and their approximations B_i are sparse. To simplify the notation, we introduce packed subgradients $\hat{g}_i \in R^{n_i}$, packed generalized Hessian matrices $\hat{G}_i \in R^{n_i \times n_i}$ and their approximations $\hat{B}_i \in R^{n_i \times n_i}$. Defining vectors $\hat{x}_i \in R^{n_i}$ as parts of vector $x \in R^n$, we can write packed quasi-Newton conditions in the form $\hat{B}_i^{k+1} \hat{s}_i^k = \hat{y}_i^k$, where $\hat{s}_i^k = \hat{x}_i^{k+1} - \hat{x}_i^k$ and $\hat{y}_i^k = \hat{g}_i^{k+1} - \hat{g}_i^k$. Packed quasi-Newton conditions imply packed quasi-Newton updates, which are used instead of dense variable metric updates.

Matrices B_i^k and subgradients g_i^k (determined from packed matrices $\hat{B}_i^k$ and packed subgradients $\hat{g}_i^k$) define matrix B^k and subgradient g^k as sums

$$B^k = \sum_{i=1}^{m} B_i^k, \quad \tilde{g}^k = \sum_{i=1}^{m} g_i^k.$$

Denoting by $\tilde{g}^k = \sum_{i=1}^m \tilde{g}_i^k$ the corresponding aggregate subgradient (see (3.14)), direction vector d^k is determined by solving the equation

$$B^k d^k = -\tilde{g}^k. \tag{3.13}$$

Furthermore, we define $w^k = -(1/2)(s^k)^T \tilde{g}^k + \tilde{\alpha}^k$. Since matrix B^k is large and sparse, we use a sparse Choleski (or Gill-Murray [8]) decomposition $B^k = L^k D^k (L^k)^T$. This decomposition is also used in the quadratic programming subproblem (3.11) instead of H^k. Thus matrix multiplications are replaced by solutions of systems with triangular matrices (back elimination). Solving (3.11) we obtain Lagrange multipliers λ_1, λ_2, λ_3. The aggregate subgradients are obtained by the formula

$$\tilde{g}_i^{k+1} = \lambda_1^k g_i^k + \lambda_2^k g_i^{k+1} + \lambda_3^k \tilde{g}_i^k, \quad 1 \le i \le m. \tag{3.14}$$

Packed matrices $\hat{B}_i^k$, $1 \le i \le m$, are updated by packed variable metric updates. We use the packed BFGS update

$$\hat{B}_i^{k+1} = \hat{B}_i^k + \frac{\hat{y}_i^k (\hat{y}_i^k)^T}{(\hat{s}_i^k)^T \hat{y}_i^k} - \frac{\hat{B}_i^k \hat{s}_i^k (\hat{B}_i^k \hat{s}_i^k)^T}{(\hat{s}_i^k)^T \hat{B}_i^k \hat{s}_i^k}, \quad (\hat{s}_i^k)^T \hat{y}_i^k > 0$$
$$\hat{B}_i^{k+1} = \hat{B}_i^k, \quad (\hat{s}_i^k)^T \hat{y}_i^k \le 0$$

after a descent step and symmetric rank-1 update

$$\hat{B}_i^{k+1} = \hat{B}_i^k + \frac{\hat{v}_i^k (\hat{v}_i^k)^T}{(\hat{s}_i^k)^T \hat{v}_i^k}, \quad (\hat{s}_i^k)^T \hat{v}_i^k > 0$$
$$\hat{B}_i^{k+1} = \hat{B}_i^k, \quad (\hat{s}_i^k)^T \hat{v}_i^k \le 0$$

with $\hat{v}_i^k = \hat{y}_i^k - \hat{B}_i^k \hat{s}_i^k$ after a zero step.

Methods for large-scale nonsmooth optimization were tested by using a set of 22 test problems with 50, 500 and 1000 variables implemented in subroutine `TEST15`, which can be downloaded from `www.cs.cas.cz/~luksan/test.html`. The results are presented in Table 4, where N is the number of variables, MET is the method used (PBM - the proximal bundle method, NVM - the nonsmooth variable metric method, SNVM - the shifted limited-memory nonsmooth variable metric method, PNVM - the partitioned nonsmooth variable metric method), NIT is the total number of iterations, NEV is the total number of function and subgradient evaluations, NF is the number of failures for a given set (i.e., the number of problems which were not successfully solved) and TIME is the total computational time in seconds.

N	MET	NIT	NEV	NF	TIME
50	PBM	55960	56854	3	29.61
	NVM	28325	28405	-	4.06
	SNVM	42243	42326	-	4.50
	PNVM	13421	13557	-	2.53
500	NVM	91832	91973	2	1281.74
	SNVM	88389	88409	3	119.91
	PNVM	15294	15369	-	32.74
1000	PNVM	14951	14976	-	166.13

Table 4

The results presented in this table imply the following conclusions:

- Nonsmooth variable metric method NVM is more efficient than proximal bundle method for small-size partially separable sums of absolute values.
- Partitioned nonsmooth variable metric method PNVM is very robust, much more efficient than other methods used for solving our set of test problems.

3.4 Variable metric methods for partially separable minimax problems

Consider functions of the form

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

where $f_i(x)$, $1 \le i \le m$ (m is usually large), are nonsmooth functions depending on a small number of variables (n_i, say). Let $F(x) = f_i(x)$ for some $1 \le i \le m$. Then any subgradient of $f_i(x)$ is a subgradient of $F(x)$. Thus we can easily found a sparse subgradient $g(x) = g_i(x)$ (containing only n_i nonzero elements) at an arbitrary point $x \in R^n$ and the corresponding quadratic programming subproblem: minimize

$$\frac{1}{2} d^T G^k d + v$$

subject to

$$-\alpha_j^k + d^T g^j \le v, \quad j \in \mathcal{J}_k, \quad -\alpha_a^k + d^T g_a^k \le v$$

has sparse constraints (note that aggregate subgradient g_a^k need not be sparse, which implies that the constraint Jacobian matrix can have one dense row). If $G^k = \sigma^k I$, we obtain a sparse quadratic programming subproblem. Thus having an efficient sparse QP solver, we can use the proximal bundle method.

Let

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

where $f_i(x)$, $1 \le i \le m$, are smooth functions depending on a small number of variables. Then minimization of F is equivalent to the sparse nonlinear programming problem with $n + 1$ variables $x \in R^n$, $z \in R$: Minimize z subject to

$$-z \le f_i(x) \le z, \quad 1 \le i \le m.$$

This problem can be solved by an arbitrary nonlinear programming method utilizing sparsity (SQP, interior point, nonsmooth equation). A special form of this problem allows us to use some simplifications in comparison with general problems. Choosing a suitable initial value of z we obtain a feasible starting point. Moreover, function $F(x)$ is an ideal merit function for the above problem. Now we are developing computer codes for an implementation of this approach.

4 Hybrid methods for large-scale nonlinear least squares

Consider functions of the form

$$F(x) = \frac{1}{2}\sum_{i=1}^{m} f_i^2(x) = \frac{1}{2} f^T(x) f(x)$$

(sum of squares), where $f_i(x)$, $1 \le i \le m$ (m is usually large), are smooth functions depending on a small number of variables (n_i, say). In this case, the Jacobian matrix $J(x) = [J_{ij}(x)] = [\partial f_i(x)/\partial x_j]$ is sparse. Using the Jacobian matrix, we can express gradient $g(x)$ and Hessian matrix $G(x)$ in the form $g(x) = J^T(x)f(x)$ and

$$G(x) = \sum_{i=1}^{m} \left(g_i(x) g_i^T(x) + f_i(x) G_i(x) \right) = J^T(x) J(x) + C(x)$$

($G_i(x)$ are Hessian matrices of $f_i(x)$, $1 \le i \le m$).

The most known Gauss-Newton method uses matrix $B(x) = J^T(x)J(x)$ instead of the Hessian matrix $G(x) = J^T(x)J(x)+C(x)$ (i.e., it omits the second order information contained in $C(x)$). We assume that matrix $J^T(x)J(x)$ is sparse (then also $C(x)$ is sparse). Matrix $J^T(x)J(x)$ is frequently ill-conditioned (even singular), thus the Gauss-Newton method requires a trust-region realization. If the minimum value $F(x^*)$ is large (large residual problem), then the Gauss-Newton method can be inefficient. Therefore, modifications based on variable metric updates has been developed. The following theorem is proved in [1].

Theorem 5. If $F_k \to 0$ Q-superlinearly, then $(F_k - F_{k+1})/F_k \to 1$. If $F_k \to F^* > 0$, then $(F_k - F_{k+1})/F_k \to 0$.

Theorem 5 implies the following philosophy of hybrid Gauss-Newton methods with second order corrections. Direction vector d is obtained by a trust-region strategy using the quadratic model $(1/2)d^T B d + f^T J d$ and the constraint $\|d\| \le \Delta$. Then $x_+ = x + d$, $F_+ = F(x_+)$ and $J_+ = J(x_+)$. If $F - F_+ > \underline{\vartheta} F$, then $B_+ = J_+^T J_+$ (Gauss-Newton method). If $F - F_+ \le \underline{\vartheta} F$, then $B_+ = J_+^T J_+ + C_+$, where C_+ is an approximation of the second order term. Usually $\underline{\vartheta} \approx 10^{-4}$.

For medium-size problems with dense matrices, matrix C is usually obtained by variable metric updates [1] [3], which are unsuitable in the large-scale case. Fortunately, simple corrections utilizing sparsity considerably increase efficiency of the Gauss-Newton method. We shortly describe two hybrid methods proposed in [17].

- Gauss-Newton method with the Newton corrections. In the first iteration we use matrix $B = J^T J$. In the subsequent iterations, we set

$$B_+ = J_+^T J_+ \quad , \quad F - F_+ > \underline{\vartheta} F,$$
$$B_+ = J_+^T J_+ + \sum_{k=1}^{m} f_k^+ G_k^+ \quad , \quad F - F_+ \le \underline{\vartheta} F,$$

where $f_k^+ = f_k(x_+)$, $G_k^+ \approx G_k(x_+)$, $1 \leq k \leq m$, (G_k^+ is a difference approximation of the Hessian matrix $G_k(x_+)$).

- Gauss-Newton method with the Marwil corrections. In the first iteration we use matrix $B = J^T J$. In the subsequent iterations, we set

$$\begin{aligned} B_+ &= J_+^T J_+ \quad , \quad F - F_+ > \underline{\vartheta} F, \\ B_+ &= \mathcal{P}_S \mathcal{P}_{QG}(J_+^T J_+) \quad , \quad F - F_+ \leq \underline{\vartheta} F, \end{aligned}$$

  where

$$\mathcal{P}_S W = (W + W^T)/2$$

  for a given square matrix W and

$$\mathcal{P}_{QG} M = \mathcal{P}_G(M + us^T).$$

  for a given symmetric positive semidefinite matrix M. Here $u \in R^n$ solves linear system $Du = y - Ms$ with diagonal matrix D such that

$$D_{ii} = \sum_{M_{ij} \neq 0} s_j^2$$

  and

$$\begin{aligned} (\mathcal{P}_G W)_{ij} &= W_{ij}, \quad (J^T J)_{ij} \neq 0, \\ (\mathcal{P}_G W)_{ij} &= 0, \qquad (J^T J)_{ij} = 0 \end{aligned}$$

  ($\mathcal{P}_G$ is the so-called gangster operator).

Methods for large-scale nonlinear least squares were tested by using a set of 52 test problems with 1000 variables implemented in subroutines `TEST15` and `TEST18`, which can be downloaded from `www.cs.cas.cz/~luksan/test.html`. The results are presented in Table 5, where SL is the strategy for step-length selection (MS - the optimum trust-region step of Moré and Sorensen [23], DL - the dog-leg strategy of Powell [25], LS - the standard line-search procedure), MET is the method used (GN - the Gauss-Newton method, GNN - the Gauss-Newton method with the Newton corrections, GNM - the Gauss-Newton method with the Marwil corrections, DN - the discrete Newton method, where the second order derivatives are approximated by differences, PVM - the partitioned variable metric method), NIT is the total number of iterations, NEV is the total number of function evaluations, NF is the number of failures for a given set (i.e., the number of problems which were not successfully solved) and TIME is the total computational time in seconds.

SL	MET	NIT	NEV	NF	TIME
MS	GN	8542	8929	1	72.00
	GNN	5499	5801	-	51.94
	GNM	6434	6801	-	62.88
	DN	7804	52398	1	202.07
DL	GN	9244	9602	-	38.84
	GNN	7767	8216	-	35.68
	GNM	6851	7029	-	25.87
	DN	10326	91181	-	171.98
LS	PVM	12093	16285	1	99.17

Table 5

The results presented in this table imply the following conclusions:

- Modifications of the Gauss-Newton method implemented with the trust-region strategy are very robust for our set of test problems, much better than discrete versions of the Newton method and more efficient than partitioned variable metric methods.
- The Newton corrections or the Marwil variable metric updates improve the efficiency of the Gauss-Newton method especially if direct methods for solving trust-region subproblems are used. Hybrid methods GNN and GNM are shown to be the most efficient methods for solving our set of test problems.

5 Methods for solving large-scale trust-region subproblems

Trust-region methods can be used when the Hessian matrix (or its approximation) is known. These methods are very convenient when this matrix is indefinite, ill-conditioned or singular. This situation often arises in connection with the Newton method for general objective function (indefiniteness) or with the Gauss-Newton method for nonlinear least-squares (near-singularity).

The crucial part of each trust region method is the direction determination. We restrict our attention to problems with large dimensions. To simplify the notation, we omit index k and use symbol $\succeq$ for ordering by positive semidefiniteness. Let

$$Q(d) = \frac{1}{2} d^T B d + g^T d.$$

We seek a direction vector $d \in R^n$ in such a way that

$$\|d\| \leq \Delta, \tag{5.1}$$

$$\|d\| < \Delta \Rightarrow \|Bd + g\| \leq \omega \|g\| \tag{5.2}$$

with $0 \leq \omega < 1$ and

$$Q(d) \geq \sigma \|g\| \min\left(\Delta, \frac{\|g\|}{\|B\|}\right) \tag{5.3}$$

with $0 < \sigma \leq 1/2$. It can be shown [26] that conditions (5.1)–(5.3) guarantee that the trust-region method is globally convergent if matrices B are uniformly bounded (or the sum of the reciprocal values of its norms is equal to infinity). There are various commonly known methods for computing direction vectors satisfying conditions (5.1)-(5.3) which we now shortly mention.

The most sophisticated method is based on the computation of the optimal locally constrained step. In this case, vector $d \in R^n$ is obtained by solving subproblem

$$\text{minimize} \quad Q(d) = \frac{1}{2} d^T B d + g^T d \quad \text{subject to} \quad \|d\| \leq \Delta. \tag{5.4}$$

Necessary and sufficient conditions for this solution are

$$\|d\| \leq \Delta, \quad (B + \lambda I)d + g = 0, \quad B + \lambda I \succeq 0, \quad \lambda \geq 0, \quad \lambda(\Delta - \|d\|) = 0. \tag{5.5}$$

The Moré-Sorensen method [23] is based on solving nonlinear equation $1/\|d(\lambda)\| = 1/\Delta$ with $(B + \lambda I)d(\lambda) + g = 0$ by the Newton method using the sparse Choleski decomposition of $B + \lambda I$. This method is very robust but requires 2-3 Choleski decompositions per iteration.

Simpler methods are based on minimization of $Q(d)$ on the two-dimensional subspace containing Cauchy step $d_C = -(g^T g / g^T B g) g$ and Newton step $d_N = -B^{-1} g$. The most popular is the dog-leg method [25], [4], where $d = d_N$ if $d_N \leq \Delta$ and $d = (\Delta / \|d_C\|) d_C$ if $\|d_C\| \geq \Delta$. In the remaining case, d is a convex combination of d_C and d_N such that $\|d\| = \Delta$. This method requires only one Choleski decomposition per iteration.

If B is not sufficiently sparse, then the sparse Choleski decomposition of B is expensive. In this case, iterative methods based on conjugate gradients are more suitable. Steihaug [27] and Toint [28] proposed a method based on the fact that $Q(d_{k+1}) < Q(d_k)$ and $\|d_{k+1}\| > \|d_k\|$ hold in the subsequent CG iterations if CG coefficients are positive. We either obtain an unconstrained solution with a sufficient precision or stop on the trust-region boundary if a negative curvature is indicated or the trust-region is left. This method is very efficient in practice especially when suitable preconditioning is used. Note that $\|d_{k+1}\|_C > \|d_k\|_C$ (where $\|d_k\|_C^2 = d_k^T C d_k$) holds instead of $\|d_{k+1}\| > \|d_k\|$ if preconditioner C (symmetric and positive definite) is used. Thus the solution on the trust-region boundary obtained by the preconditioned CG method can be farther from the optimal locally constrained step than the solution obtained without preconditioning. This insufficiency is usually compensated by the rapid convergence of the preconditioned CG method.

The CG steps can be combined with Newton step d_N in the multiple dog-leg method [27], [16]. Let $k \ll n$ (usually $k = 5$) and d_k be a vector obtained after k CG steps of the Steihaug-Toint method. If $\|d_k\| < \Delta$, we use d_k instead of $d_C = d_1$ in the dog-leg method.

The solution on the trust-region boundary obtained by the Steihaug-Toint method can be rather far from the optimal solution. This insufficiency can be overcame by using the Lanczos process [10]. Initially, the conjugate gradient algorithm is used as in the Steihaug-Toint method. At the same time, the Lanczos tridiagonal matrix is

constructed from the CG coefficients. If a negative curvature is indicated or the trust-region is left, we turn to the Lanczos process. In this case, $d = Z\tilde{d}$, where $\tilde{d}$ is obtained by minimizing quadratic function

$$\frac{1}{2}\tilde{d}^T T\tilde{d} + \|g\|e_1^T\tilde{d}$$

subject to $\|\tilde{d}\| \leq \Delta$. Here $T = Z^T BZ$ (with $Z^T Z = I$) is the Lanczos tridiagonal matrix and e_1 is the first column of the unit matrix. This method cannot be successfully preconditioned, since preconditioning changes the original trust-region subproblem to $\|d\|_C \leq \Delta$ to $\|d\|_C \leq \Delta$, where C changes in each major iteration and can be ill-conditioned.

To overcome the insufficiency of the previous method, the Lanczos process can be combined with the Steihaug-Toint method. The shifted Steihaug-Toint method proposed in [18] consists of three steps:

- Let $m \ll n$ (usually $m = 5$). Determine tridiagonal matrix T of order m by m steps of the (unpreconditioned) Lanczos method applied to matrix B with the initial vector g.
- Solve subproblem

$$\text{minimize} \quad \frac{1}{2}\tilde{d}^T T\tilde{d} + \|g\|e_1^T\tilde{d} \quad \text{subject to} \quad \|\tilde{d}\| \leq \Delta \tag{5.6}$$

  using the method of Moré and Sorensen to obtain Lagrange multiplier $\tilde{\lambda}$.
- Apply the (preconditioned) Steihaug-Toint method to subproblem

$$\text{minimize} \quad \frac{1}{2}d^T(B + \tilde{\lambda}I)d + g^T d \quad \text{subject to} \quad \|d\| \leq \Delta \tag{5.7}$$

  to obtain direction vector $d = d(\tilde{\lambda})$.

The following theorem is proved in [18].

Theorem 6. Let $\tilde{\lambda}$ be the Lagrange multiplier of the small-size subproblem (5.6) and λ be the Lagrange multiplier obtained by the Moré-Sorensen method applied to the original problem. Then $0 \leq \tilde{\lambda} \leq \lambda$.

As a consequence of Theorem 6, one has that $\lambda = 0$ implies $\tilde{\lambda} = 0$ so that $\|d\| < \Delta$ implies $\tilde{\lambda} = 0$. Thus the shifted Steihaug-Toint method reduces to the standard one in this case. At the same time, if B is positive definite and $\tilde{\lambda} > 0$, then one has $\Delta \leq \|(B + \tilde{\lambda}I)^{-1}g\| < \|B^{-1}g\|$. Thus the unconstrained minimizer of the shifted quadratic function (5.7) is closer to the trust-region boundary than the unconstrained minimizer of the original quadratic function (5.4) and we can expect that $d(\tilde{\lambda})$ is closer to the optimal locally constrained step than d. Finally, if $\tilde{\lambda} > 0$, then matrix $B + \tilde{\lambda}I$ is better conditioned than B and we can expect that the shifted Steihaug-Toint method will converge more rapidly than the original one.

Methods for solving large-scale trust-region subproblems were tested by using a set of 22 sparse test problems with 1000 and 5000 variables implemented in subroutine `TEST14`, which can be downloaded from `www.cs.cas.cz/~luksan/test.html`. The results are presented in Table 6, where N is the number of variables, MET is the method used (MS - the optimum trust-region step of Moré and Sorensen [23], DL - the dog-leg strategy of Powell [25], MDL - the multiple dog-leg strategy [16] with $m = 5$, ST - the basic Steihaug-Toint method, GLRT - the method of Gould, Lucidi, Roma and Toint [10] based on the Lanczos process, PST - the preconditioned Steihaug-Toint method (with the incomplete Choleski preconditioner), PSST - the preconditioned shifted Steihaug-Toint method [18] with $m = 5$), NIT is the total number of iterations, NEV is the total number of function evaluations, NCG is the total number of CG iterations and TIME is the total computational time in seconds.

N	MET	NIT	NEV	NCG	TIME
1000	MS	1918	1955	-	4.65
	DL	2515	2716	-	4.42
	MDL	2292	2456	12203	4.61
	ST	3329	3784	53573	8.20
	GLRT	3107	3444	55632	8.53
	PST	2631	2823	910	5.14
	PSST	1999	2046	1161	4.25
5000	MS	8391	8566	-	2:02.44
	DL	9657	10133	-	1:55.77
	MDL	8938	9276	47236	2:02.84
	ST	16894	19163	358111	6:04:42
	GLRT	14679	16383	366695	6:41.45
	PST	10600	11271	3767	2:25.42
	PSST	8347	8454	4329	1:48.87

Table 6

The results presented in this table imply the following conclusions:

- Direct methods MS and DL based on the sparse Choleski decomposition are very efficient for our set of test problems. Iterative methods require a suitable preconditioning.
- The Moré-Sorensen strategy MS gives the best approximation of the optimum locally constrained step and decreases the number of the major iterations.
- New strategy PSST can be efficiently preconditioned. It gives a relatively good approximation of the optimum locally constrained step. Method PSST is the most efficient method for solving our set of test problems.

References

[1] M.Al-Baali, R.Fletcher: Variational Methods for Nonlinear Least Squares. Journal of Optimization Theory and Applications 36 (1985) 405-421.

[2] R.H.Byrd, J.Nocedal, R.B.Schnabel: Representation of quasi-Newton matrices and their use in limited memory methods. Math. Programming 63 (1994) 129-156.

[3] J.E.Dennis, D.Gay, R.E.Welsch: An adaptive nonlinear least squares algorithm. ACM Trans. on Math. Software 7 (1981) 348-368.

[4] J.E.Dennis, H.H.W.Mei: An unconstrained optimization algorithm which uses function and gradient values. Report No. TR 75-246, 1975.

[5] N.I.M.Gould, S.Lucidi, M.Roma, P.L.Toint: Solving the trust-region subproblem using the Lanczos method. Report No. RAL-TR-97-028, 1997.

[6] P.E.Gill, M.W.Leonard: Reduced-Hessian quasi-Newton methods for unconstrained optimization. SIAM J. on Optimization 12 (2001) 209-237.

[7] P.E.Gill, M.W.Leonard: Limited-memory reduced-Hessian methods for large-scale unconstrained optimization. SIAM J. Optimization 14 (2003) 380-401.

[8] P.E.Gill, W.Murray: Newton type methods for unconstrained and linearly constrained optimization. Math. Programming 7 (1974) 311-350.

[9] P.E.Gill, W.Murray, M.A.Saunders: Methods for computing and modifying LDV factors of a matrix. Math. Comput. 29 (1975) 1051-1077.

[10] N.I.M.Gould, S.Lucidi, M.Roma, P.L.Toint: Solving the trust-region subproblem using the Lanczos method. Report No. RAL-TR-97-028, 1997.

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

[12] M.Haarala, K.Mietinen, M.M.Mäkelä: Large-scale nonsmooth optimization: New variable-metric bundle algorithm with limited memory. Report No. B 6/2003, Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä 2003.

[13] M.Haarala, K.Mietinen, M.M.Mäkelä: Limited memory bundle method for large-scale nonsmooth optimization: Convergence analysis. Report No. B 12/2003, Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä 2003.

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

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

[16] L.Lukšan: Combined trust region methods for nonlinear least squares. Kybernetika 32 (1996) 121-138.

[17] L.Lukšan: Hybrid methods for large sparse nonlinear least squares. J. Optimizaton Theory and Applications 89 (1996) 575-595.

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

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

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

[21] L.Lukšan, J.Vlček: Variable metric method for minimization of partially separable nonsmooth functions. Report V-916, Prague, ICS AS CR, 2004.

[22] M.M.Mäkelä, P.Neittaanmäki: Nonsmooth Optimization. World Scientific Publishing Co., London, 1992.

[23] J.J.Moré, D.C.Sorensen: Computing a trust region step. SIAM Journal on Scientific and Statistical Computations 4 (1983) 553-572.

[24] J. Nocedal: Updating quasi-Newton matrices with limited storage. Math. Comp. 35 (1980) 773-782.

[25] M.J.D.Powell: A new algorithm for unconstrained optimization. In: Nonlinear Programming (J.B.Rosen O.L.Mangasarian, K.Ritter, eds.) Academic Press, London 1970.

[26] M.J.D.Powell: On the global convergence of trust region algorithms for unconstrained optimization. Mathematical Programming 29 (1984) 297-303.

[27] T.Steihaug: The conjugate gradient method and trust regions in large-scale optimization. SIAM Journal on Numerical Analysis 20 (1983) 626-637.

[28] P.L.Toint: Towards an efficient sparsity exploiting Newton method for minimization. In: Sparse Matrices and Their Uses (I.S.Duff, ed.), Academic Press, London 1981, 57-88.

[29] J.Vlček, L.Lukšan: Globally convergent variable metric method for nonconvex nondifferentiable unconstrained minimization. Journal of Optimization Theory and Applications, Vol.111, 2001, pp.407-430.

[30] J.Vlček, L.Lukšan: New variable metric methods for unconstrained minimization covering the large-scale case. Report V-876, Prague, ICS AS CR, 2002.

[31] J.Vlček, L.Lukšan: Additional properties of shifted variable metric methods. Report V-899, Prague, ICS AS CR, 2004.