Gauss-Newton-based BFGS method with filter for unconstrained minimization

Nataša Krejić* Zorana Lužanin † Irena Stojkovska‡

June 9, 2008

Abstract

One class of the lately developed methods for solving optimization problems are filter methods. In this paper we attached a multidimensional filter to the Gauss-Newton-based BFGS method given by Li and Fukushima [*SIAM J.Numer.Anal.*, Vol.37, No.1 (1999), pp.152-172] in order to reduce the number of backtracking steps. The proposed filter method for unconstrained minimization problems converges globally under the standard assumptions. It can also be successfully used in solving systems of symmetric nonlinear equations. Numerical results show reasonably good performance of the proposed algorithm.

Key words. BFGS method, filter methods, global convergence, unconstrained minimization

AMS subject classification. 65H10, 90C53

1 Introduction

Consider the following problem

$$\min_{x\in\mathbb{R}^n} f(x), \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is two times continuously differentiable function.

There are various methods for solving the unconstrained minimization problem (1), see [8, 17]. One of the most exploited are quasi-Newton methods, firstly proposed by Broyden [3] in 1965. They are defined by the iterative formula

$$x_{k+1} = x_k + \lambda_k p_k,$$

*Department of Mathematics and Informatics, Faculty of Science, University of Novi Sad, Trg Dositeja Obradovića 4, 21000 Novi Sad, Serbia, e-mail: natasak@uns.ns.ac.yu

†Department of Mathematics and Informatics, Faculty of Science, University of Novi Sad, Trg Dositeja Obradovića 4, 21000 Novi Sad, Serbia, e-mail: zorana@im.ns.ac.yu

‡Department of Mathematics, Faculty of Natural Sciences and Mathematics, St. Cyril and Methodius University, Gazi Baba b.b., 1000 Skopje, Macedonia, e-mail: irenatra@iunona.pmf.ukim.edu.mk

where $\lambda_k > 0$ is the step length that is updated by line search and backtracking procedure and $p_k = -B_k^{-1}\nabla f(x_k)$ is the search quasi-Newton direction, where $\nabla f(x)$ is the gradient mapping of $f(x)$. In quasi-Newton methods B_k is an approximation of the Hessian $\nabla^2 f(x_k)$ and it is updated at every iteration with some low-rank matrix. One of the well known update formula is the BFGS formula given by

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k},$$

where $s_k = x_{k+1} - x_k$ and $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$. The BFGS update formula possesses nice properties that are very useful for establishing convergence results, see [4]. For better understanding of quasi-Newton methods, we refer to [2, 7, 16].

In [15], Li and Fukushima proposed a new modified BFGS method called Gauss-Newton-based BFGS method for symmetric nonlinear equations

$$g(x) = 0, \tag{2}$$

where the mapping $g : \mathbb{R}^n \to \mathbb{R}^n$ is differentiable and the Jacobian $\nabla g(x)$ is symmetric for all $x \in \mathbb{R}^n$. By introducing a new line search technique and modifying the BFGS update formula, under suitable conditions, they proved global and superlinear convergence of their method.

If we denote by $g : \mathbb{R}^n \to \mathbb{R}^n$, the gradient mapping of the function $f(x)$ from (1), then $g(x)$ is differentiable and the Hessian $\nabla^2 f(x)$ is symmetric for all $x \in \mathbb{R}^n$. The first order optimality conditions for (1) are given by (2) and therefore the method from [15] is applicable for solving (1).

Like many other quasi-Newton methods the Gauss-Newton-based BFGS method [15] uses a backtracking procedure in determining the step length. A line search procedure can result in a very small step lengths. In order to avoid these small steps as well as to reduce the number of backtracking procedures, in this paper we associate a filter to the Gauss-Newton-based BFGS method.

Filter methods, one of the latest developments in global optimization algorithms, were firstly proposed by Fletcher in 1996, [9]. First filter methods are a kind of alternative to penalty functions used in constrained nonlinear programming optimization algorithms, see [5, 11, 14, 18]. In these methods, filter functions penalize the constraints violations less restrictively then penalty functions. The main purpose of filters is allowing the full Newton step to be taken more often and thus inducing global convergence of the method. Filter methods are extended to solving nonlinear equations, see [6, 12], unconstrained optimization, [13]. For detailed reading about filter methods developments we refer to [10].

In this paper we add a filter to the Gauss-Newton-based BFGS method [15] in order to solve the unconstrained minimization problem (1). Our filter method has global convergence property and in some test problems shows better performance then the method without filter. The filter in our paper is inspired by the multidimensional filter in [6, 12] although there are minor differences.

In Section 2 we present the Gauss-Newton-based BFGS method [15], introduce the multidimensional filter and state our Gauss-Newton-based BFGS method with filter. In Section 3 we proved the global convergence of our method under the set of standard assumptions. Some numerical results are reported in Section 4.

2 Algorithm

Let us first briefly explain the main properties of the Gauss-Newton-based BFGS method [15] for solving problem (1). That method will be used in combination with a filter in this paper.

Given the gradient mapping g and current iteration $x_k = x_{k-1} + \lambda_{k-1} p_{k-1}$, the search direction p_k is obtained from the following linear system

$$B_k p + \lambda_{k-1}^{-1}(g(x_k + \lambda_{k-1} g(x_k)) - g(x_k)) = 0. \tag{3}$$

The line search is applied if the inequality

$$||g(x_k + \lambda_k p_k)|| \leq \rho ||g(x_k)|| \tag{4}$$

is not satisfied for some fixed $\rho \in [0, 1)$. The line search rule is governed by a positive sequence $\{\varepsilon_k\}$ satisfying $\sum_k \varepsilon_k < \infty$. For the chosen sequence $\{\varepsilon_k\}$ and fixed parameters $\sigma_1, \sigma_2 > 0$ we take $x_{k+1} = x_k + \lambda_k p_k$ as a new iterate if

$$||g(x_k + \lambda_k p_k)||^2 - ||g(x_k)||^2 \leq -\sigma_1 ||\lambda_k g(x_k)||^2 - \sigma_2 ||\lambda_k p_k||^2 + \varepsilon_k ||g(x_k)||^2. \tag{5}$$

Otherwise the step size is decreased, $\lambda_k := r\lambda_k,\ 0 < r < 1$ until (5) is satisfied. The acceptance rule (5) starts with $\lambda_k = 1$ and is well defined since the inequality has to be satisfied for $\lambda_k > 0$ small enough due to the presence of $\varepsilon_k > 0$ at the right-hand side of (5). After determining x_{k+1} the new matrix B_{k+1} is obtained by (6), where

$$\begin{aligned} s_k &= x_{k+1} - x_k = \lambda_k p_k, \\ \delta_k &= g(x_{k+1}) - g(x_k), \\ y_k &= g(x_k + \delta_k) - g(x_k), \\ B_{k+1} &= B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}. \end{aligned} \tag{6}$$

The usual safe guard condition $y_k^T s_k > 0$ is applied i.e. if $y_k^T s_k \leq 0$ then $B_{k+1} = B_k$.

In computational implementation of the algorithm presented in Section 4 we used BFGS approximation of the inverse Jacobian, $H_k = B_k^{-1}$ and therefore determine the search direction as

$$p_k = -H_k \lambda_{k-1}^{-1}(g(x_k + \lambda_{k-1} g(x_k)) - g(x_k)) \tag{7}$$

with

$$H_{k+1} = (I - \frac{s_k y_k^T}{y_k^T s_k}) H_k (I - \frac{y_k s_k^T}{y_k^T s_k}) + \frac{s_k s_k^T}{y_k^T s_k}. \tag{8}$$

The Gauss-Newton based BFGS is globally convergent under the set of standard assumptions, [15]. The same set of assumption is used in this paper and is listed as A1-A3 in Section 3.

Let us now introduce the multidimensional filter we will use in combination with the Gauss-Newton based BFGS from [15]. Our filter is defined similarly as in [6, 12]. The equations (2) are partitioned into m sets $\{g_i(x)\}_{i\in I_j}$, $j = 1, ..., m$, with the property $\{1, ..., n\} = I_1 \cup ... \cup I_m$, $I_j \cap I_k = \varnothing$, $j \neq k$ and the *filter functions* are defined as

$$\phi_j(x) \stackrel{def}{=} ||g_{I_j}(x)|| \quad \text{for} \quad j = 1, ..., m \tag{9}$$

where $||\cdot||$ is the Euclidean norm and g_{I_j} is the vector whose components are the components of g indexed by I_j. With this notation x^* satisfies the optimality conditions of (1) if and only if $\phi_j(x^*) = 0$ for all $j = 1, ..., m$. The following abbreviations will be used

$$\phi(x) \stackrel{def}{=} (\phi_1(x), ..., \phi_m(x))^T,\ \phi_k \stackrel{def}{=} \phi(x_k) \text{ and } \phi_{j,k} \stackrel{def}{=} \phi_j(x_k).$$

A *filter* is a list $\mathcal{F}$ of m-tuples of the form $(\phi_{1,k}(x), \phi_{2,k}(x), ..., \phi_{m,k}(x))$ such that

$$\phi_{j,k} < \phi_{j,l} \text{ for at least one } j \in \{1, ..., m\} \text{ and for all } k \neq l.$$

To understand the meaning and usage of the filter a concept of domination is introduced. A point x_1 *dominates* a point x_2 whenever

$$\phi_j(x_1) \leq \phi_j(x_2) \text{ for all } j = 1, ..., m.$$

Therefore we say that the filter keeps all iterates that are not dominated by other iterates in the filter. In this work, we use a filter to construct an additional acceptability condition for a new trial iterate $x_k^+ = x_k + s_k$. This condition is slightly different than the one in [12]. The first difference is the function δ_1 in (10), where we use max function while in [12] min is considered. The procedure for removing points from the filter is also different. The reason for these changes is empirical since we realized that the algorithm is more efficient with the rule proposed in this paper.

We say that a new trial point x_k^+ is *acceptable for the filter* $\mathcal{F}$ if

$$\forall \phi_l \in \mathcal{F} \quad \exists j \in \{1, ..., m\} \quad \phi_j(x_k^+) < \phi_{j,l} - \gamma_\phi \delta_1(||\phi_l||, ||\phi_k^+||), \tag{10}$$

where $\gamma_\phi \in (0, 1)$ is a small positive constant and

$$\delta_1(||\phi_l||, ||\phi_k^+||) = \max\{||\phi_l||, ||\phi_k^+||\}.$$

When a trial point is acceptable for the filter, we *add the trial point to the filter* immediately (again different from [12]). In other words add the m-tuple $\phi_k^+ = \phi(x_k^+) = (\phi_1(x_k^+), ..., \phi_m(x_k^+))^T$ to the filter $\mathcal{F}$.

When we add a new trial point x_k^+ to the filter, we remove all points from the filter that are dominated by the trial point x_k^+, which means that we remove m-tuples $(\phi_{1,k_i}, ..., \phi_{m,k_i})^T \in \mathcal{F} \setminus \{\phi_k^+\}$ from the filter if

$$\phi_j(x_k^+) \leq \phi_{j,k_i} \quad j = 1, ..., m.$$

Every trial point which is acceptable for the filter is taken as a new iterate.

Our implementation of the multidimensional filter in Gauss-Newton based BFGS with filter method is done in such a way that the backtracking $\lambda_k := r\lambda_k$ is enforced only if a trial point x_k^+ is not acceptable to the filter. The proposed filter algorithm gives its best performance in the case when $m = n$ and $I_j = \{j\}$ - similarly as it is discussed in [10]). In this case we will have $\phi_j(x) = |g_j(x)|$, $\phi(x) = g(x)$, $\phi_k = g(x_k)$ and $\phi_{j,k} = |g_j(x_k)|$.

Now we are ready to state the new algorithm.

ALGORITHM GNbBFGSf. Gauss-Newton-based BFGS method with filter

Step 0. Choose an initial point $x_0 \in \mathbb{R}^n$, an initial symmetric positive definite matrix $B_0 \in \mathbb{R}^{n\times n}$, a positive sequence $\{\varepsilon_k\}$ satisfying $\sum_{k=0}^{\infty} \varepsilon_k < \infty$, and constants $r, \rho, \gamma_\phi \in (0, 1)$, $\sigma_1, \sigma_2 > 0$, $\lambda_{-1} > 0$. Let $k := 0$.

Step 1. If $g(x_k) = 0$ then Stop. Otherwise, solve the following linear equation to get p_k

$$B_k p + \lambda_{k-1}^{-1}(g(x_k + \lambda_{k-1} g(x_k)) - g(x_k)) = 0. \tag{11}$$

Take $\lambda_k = 1$ and go to Step 2.

Step 2. Let the trial point be $x_k^+ = x_k + \lambda_k p_k$. If

$$||g(x_k^+)||^2 - ||g(x_k)||^2 \leq -\sigma_1||\lambda_k g(x_k)||^2 - \sigma_2||\lambda_k p_k||^2 + \varepsilon_k||g(x_k)||^2 \tag{12}$$

then go to Step 3, else if x_k^+ is acceptable to the filter then add x_k^+ to the filter, remove all points from the filter that are dominated by x_k^+ and go to Step 3. Otherwise, put $\lambda_k := r\lambda_k$ and repeat Step 2.

Step 3. Take the next iterate $x_{k+1} = x_k^+$.

Step 4. Put

$$s_k = x_{k+1} - x_k = \lambda_k p_k,$$

$$\delta_k = g(x_{k+1}) - g(x_k),$$

$$y_k = g(x_k + \delta_k) - g(x_k).$$

If $y_k^T s_k \leq 0$, then $B_{k+1} = B_k$ and go to Step 5. Otherwise, update B_k

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} \tag{13}$$

and go to Step 5.

Step 5. Let $k := k + 1$. Go to Step 1.

At the beginning of Algorithm GNbBFGSf, the filter is initialized to be empty $\mathcal{F} = \varnothing$ or to be $\mathcal{F} = \{\phi(x) : \phi_j(x) \geq \phi_{j\max}, j = 1, ..., m\}$ for any $\phi_{j\max} > \phi_j(x_0)$, $j = 1, ..., m$ (similarly as in [18]). In practical implementations of this method, Section 4, the filter is initialized to be empty $\mathcal{F} = \varnothing$ or to be $\mathcal{F} = \{\phi(x_0)\}$.

One should notice that if there are finitely many values of ϕ_k that are added to the filter, then for all k large enough (for all $k \geq k_0$ where ϕ_{k_0} is the last m-tuple added to the filter), Algorithm GNbBFGSf is the same as the one considered in [15].

3 Convergence result

In this section, we are going to establish the global convergence of Algorithm GNbBFGSf. Some convergence results from [15] will be used. To do that we need the following assumption for the values ϕ_k that are added to the filter by Algorithm GNbBFGSf.

Assumption.

A0 There exists a constant $C > 0$ such that for all values ϕ_k that are added to the filter by Algorithm GNbBFGSf the following stands

$$||\phi_k|| \leq C.$$

Let Ω be the level set defined by

$$\Omega = \{x : ||g(x)|| \leq e^{\frac{\varepsilon}{2}} \max\{C, ||g(x_0)||\}\}, \tag{14}$$

where ε is a positive constant such that

$$\sum_{k=0}^{\infty} \varepsilon_k < \varepsilon. \tag{15}$$

Than we have the following lemma.

Lemma 3.1 *Let the assumption A0 holds and let $\{x_k\}$ be generated by Algorithm GNbBFGSf. Then $\{x_k\} \subset \Omega$.*

Proof. If for the iterate x_k there are no values $\phi_{k'}, k' \leq k$ before it that are added to the filter, then in [15] it is shown that

$$||g(x_k)|| \leq e^{\frac{\varepsilon}{2}} ||g(x_0)||. \tag{16}$$

The last inequality (16) is obtained using the following conclusion, that is: for each two consecutive iterates x_{k-1} and x_k which were not added to the filter the inequality (12) holds which means that

$$||g(x_k)|| \leq (1 + \varepsilon_{k-1})^{\frac{1}{2}} ||g(x_{k-1})||, \tag{17}$$

where ε is a constant satisfying (15).

Now let $x_{i(k)}$ be the last iterative point before x_k such that $\phi_{i(k)}$ was added to the filter. Then using (17) we have

$$\begin{aligned}
||g(x_k)|| &\leq (1+\varepsilon_{k-1})^{\frac{1}{2}}||g(x_{k-1})|| \\
&\leq (1+\varepsilon_{k-1})^{\frac{1}{2}}(1+\varepsilon_{k-2})^{\frac{1}{2}}||g(x_{k-2})|| \\
&\vdots \\
&\leq \Big(\prod_{j=i(k)}^{k-1}(1+\varepsilon_j)^{\frac{1}{2}}\Big)||g(x_{i(k)})|| \\
&\leq ||g(x_{i(k)})||\Big(\frac{1}{k-i(k)}\sum_{j=i(k)}^{k-1}(1+\varepsilon_j)\Big)^{\frac{k-i(k)}{2}} \\
&= ||g(x_{i(k)})||\Big(1+\frac{1}{k-i(k)}\sum_{j=i(k)}^{k-1}\varepsilon_j\Big)^{\frac{k-i(k)}{2}} \qquad (18)\\
&\leq ||g(x_{i(k)})||\Big(1+\frac{\varepsilon}{k-i(k)}\Big)^{\frac{k-i(k)}{2}} \\
&\leq e^{\frac{\varepsilon}{2}}||g(x_{i(k)})|| \\
&\leq e^{\frac{\varepsilon}{2}}C, \qquad (19)
\end{aligned}$$

where ε is a constant satisfying (15) and $C > 0$ is a constant from assumption A0. From (16) and (19) it implies that $\{x_k\} \subset \Omega$. $\square$

The following set of assumption is also necessary for the global convergence analysis of Algorithm GNbBFGSf.

Assumptions.

A1 g is continuously differentiable on an open set Ω_1 containing Ω.

A2 ∇g is symmetric and bounded on Ω_1 i.e. $\nabla g(x)^T = \nabla g(x)$ for every $x \in \Omega_1$ and there exists a positive constant M such that

$$||\nabla g(x)|| \leq M \quad \forall x \in \Omega_1.$$

A3 ∇g is uniformly nonsingular on Ω_1 i.e. there is a constant $m > 0$ such that

$$m||p|| \leq ||\nabla g(x)p|| \quad \forall x \in \Omega_1, p \in \mathbb{R}^n.$$

As the mapping g is the gradient of f and f is two times continuously differentiable, A1 and A2 are clearly satisfied while A3 is the standard assumption for global convergence of optimization algorithms.

In order to prove the global convergence of Algorithm GNbBFGSf we distinguish two cases. The first one is when there are finitely many values of ϕ_k that are added to the filter by Algorithm GNbBFGSf, and for that case our algorithm is the same as the Gauss-Newton based BFGS from [15]. Therefore we will not consider that case in this paper. The second case occurs when there are infinitely many values of ϕ_k that are added to the filter by Algorithm GNbBFGSf. For that case we state the following lemma.

Lemma 3.2 *Let the assumptions A0-A3 hold and assume that infinitely many values of ϕ_k are added to the filter by Algorithm GNbBFGSf. Then*

$$\lim_{k\to\infty} ||g(x_k)|| = 0. \tag{20}$$

Proof. The first part of the proof is based on [6, 12]. Let $\{k_i\}$ index the subsequence of iterations at which $\phi_{k_i} = \phi^+_{k_i-1}$ is added to the filter. Now assume that there exists a subsequence $\{k_j\} \subseteq \{k_i\}$ such that

$$||\phi_{k_j}|| \geq \varepsilon_1 \tag{21}$$

fore some constant $\varepsilon_1 > 0$. Since by the assumption we know that $\{||\phi_{k_j}||\}$ is bounded, there exists a subsequence $\{k_l\} \subseteq \{k_j\}$ such that

$$\lim_{l\to\infty} \phi_{k_l} = \phi_\infty \quad \text{with} \quad ||\phi_\infty|| \geq \varepsilon_1.$$

Moreover, by definition of $\{k_l\}$, ϕ_{k_l} is acceptable for every l, which implies in particular that for each l there exists an index $j \in \{1, ..., m\}$ such that

$$\phi_{j,k_l} - \phi_{j,k_{l-1}} < -\gamma_\phi ||\phi_{k_l}|| \tag{22}$$

as we will show now. Let us consider the following two cases.

(i) If $\phi_{k_{l-1}}$ is still in the filter, then there exists an index $j \in \{1, ..., m\}$ such that
$$\phi_{j,k_l} - \phi_{j,k_{l-1}} < -\gamma_\phi \delta_1(||\phi_{k_l}||, ||\phi_{k_{l-1}}||) \leq -\gamma_\phi ||\phi_{k_l}||.$$

(ii) If $\phi_{k_{l-1}}$ is removed by one trial point, say $\phi_{k_{l'}}$ with $k_{l-1} < k_{l'} < k_l$, and $\phi_{k_{l'}}$ is still in the filter, then there exists a $j \in \{1, ..., m\}$ such that

$$\phi_{j,k_l} - \phi_{j,k_{l'}} < -\gamma_\phi \delta_1(||\phi_{k_l}||, ||\phi_{k_{l'}}||) \leq -\gamma_\phi ||\phi_{k_l}||$$

and
$$\phi_{j,k_{l'}} \leq \phi_{j,k_{l-1}},$$

so (22) holds in this case too.

From (22), (21) and $\{k_l\} \subseteq \{k_j\}$, we deduce that

$$\phi_{j,k_l} - \phi_{j,k_{l-1}} < -\gamma_\phi \varepsilon_1,$$

but the left-hand side of it tends to zero when l tends to infinity. This leads us to the contradiction. Hence

$$\lim_{i\to\infty} ||\phi_{k_i}|| = 0. \tag{23}$$

Consider now any $l \notin \{k_i\}$ and let $k_{i(l)}$ be the last iteration before l such that $\phi_{k_{i(l)}}$ was added to the filter. Like (18) was derived, the following inequality can be derived too.

$$||g(x_l)|| \leq ||g(x_{k_{i(l)}})|| \Big(1 + \frac{1}{l - k_{i(l)}} \sum_{j=k_{i(l)}}^{l-1} \varepsilon_j \Big)^{\frac{l-k_{i(l)}}{2}}. \tag{24}$$

By the definition of $\{\varepsilon_k\}$, we know that

$$\lim_{l\to\infty} \sum_{j=k_{i(l)}}^{l-1} \varepsilon_j = 0,$$

and consequently

$$\lim_{l\to\infty} (1 + \frac{1}{l - k_{i(l)}} \sum_{j=k_{i(l)}}^{l-1} \varepsilon_j \Big)^{\frac{l-k_{i(l)}}{2}} = 1. \tag{25}$$

From previously proved (23) we have that

$$\lim_{l\to\infty} ||\phi_{k_{i(l)}}|| = 0. \tag{26}$$

The equalities (25) and (26) imply that when l tends to infinity then the right-hand side of (24) tends to zero and therefore

$$\lim_{l\to\infty} ||g(x_l)|| = 0,$$

which completes the proof. □

Lemma 3.2 and global convergence of Gauss-Newton based BFGS algorithm from [15] clearly imply the next global convergence theorem.

Theorem 3.3 *Let the assumptions A0-A3 hold. Then the sequence $\{x_k\}$ generated by Algorithm GNbBFGSf converges to the unique solution x^* of problem* (2).

Proof. If there are finitely many values of ϕ_k that are added to the filter by Algorithm GNbBFGSf, then Algorithm GNbBFGSf acts same as Gauss-Newton based BFGS algorithm from [15] after some finite number of iterative steps, and in [15] it is proved that under assumptions A1-A3 $\{||g(x_k)||\}$ converges and

$$\liminf_{k\to\infty} ||g(x_k)|| = 0. \tag{27}$$

The case when infinitely many values of ϕ_k are added to the filter by Algorithm GNbBFGSf is considered in Lemma 3.2.

So, from (27) and (20), we can say that every accumulation point of $\{x_k\}$ is a solution of (2). Since ∇g is uniformly nonsingular on Ω_1, (2) has only one solution. Since Ω is bounded, $\{x_k\} \subset \Omega$ has at least one accumulation point. Therefore $\{x_k\}$ converges to the unique solution of (2). □

4 Numerical results

In this section we report results of some numerical experiments with the proposed method and give comparison with the method considered in [15]. Problems 1, 2 and 3 are of fixed dimension, while problems 4 and 5 have can have various dimensions.

Problem 1. [6]

$$\min_{x\in\mathbb{R}^2} f(x),$$

where

$$f(x) := \frac{1}{2}((x_1^2 - x_2 - 1)^2 + ((x_1-2)^2 + (x_2-0.5)^2 - 1)^2).$$

Problem 2. [6]

$$\min_{x\in\mathbb{R}^3} f(x)$$

where

$$f(x) = \frac{1}{2}((12x_1 - x_2^2 - 4x_3 - 7)^2 + (x_1^2 + 10x_2 - x_3 - 11)^2 + (x_2^2 + 10x_3 - 8)^2).$$

Problem 3. [1] Wood's Function (WF)

$$\min_{x\in\mathbb{R}^4} f(x)$$

where

$$\begin{aligned} f(x) &= 100(x_2 - x_1^2)^2 + (1-x_1)^2 + 90(x_4 - x_3^2)^2 + (1-x_3)^2 + \\ &+10.1((x_2-1)^2 + (x_4-1)^2) + 19.8(x_2-1)(x_4-1)\}. \end{aligned}$$

Problem 4. [1] Cosine Mixture Problem (CM)

$$\min_{x\in\mathbb{R}^n} f(x)$$

where

$$f(x) = \sum_{i=1}^{n} x_i^2 - 0.1\sum_{i=1}^{n}\cos(5\pi x_i).$$

Problem 5. [1] Rosenbrock Problem (RB)

$$\min_{x\in\mathbb{R}^n} f(x)$$

where

$$f(x) = \sum_{i=1}^{n-1}(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2).$$

Experiments were done skipping Step 2 in the Gauss-Newton based BFGS method from [15] i.e. skipping the monotone decrease acceptance condition (4) and with updating the approximation the inverse Jacobian H_k. The parameters were chosen to be $r = 0.1$, $\sigma_1 = \sigma_2 = 10^{-5}$, $\lambda_{-1} = 0.01$, $\varepsilon_k = k^{-2}$ and $\gamma_\phi = 0.5$, and the initial matrix H_0 was set to be the identity matrix. Test were performed in Matlab. The iteration procedure was stopped when the condition $||g(x_k)|| \leq \text{eps}^{1/3}$ with $\text{eps} = 2^{-52}$ was satisfied.

Three methods were tested. In Table 1 and Table 2 the following abbreviations are used: Alg.1 is the Gauss-Newton-based BFGS method from [15], Alg.2(I) is the Gauss-Newton-based BFGS method with filter described in Algorithm GNbBFGSf with the initialization of the filter $\mathcal{F} = \{\phi(x_0)\}$ and Alg.2(II) is the Gauss-Newton-based BFGS method with filter described in Algorithm GNbBFGSf and the initialization $\mathcal{F} = \varnothing$.

Table 1 shows the initial point x_0, number of iteration (iter), number of calculations of the function g (gcalc) and number of steps where the filter was used (ftr) needed to solve above problems. Table 2 shows the values of the last iterative points.

The overall performance of all algorithms was good. It is clear that comparison of Alg.1 and Alg. 2 depends on an initial point. From Table 1 we can see that in some cases Alg.1 still has better performance: Problem 1 - the second initial point, Problem 2 - the first initial point, Problem 3 - the second initial point, Problem 4 - the second initial point, Problem 5 - the first and third initial point. But in the remaining cases one of the Alg. 2 or both of them are superior to Alg. 1. This shows that the usage of filter is justified and that it can reduce the number of iterations and consequently the number of calculations of the function g. In some cases the decrease was quite significant, Problem 3 - the first initial point, Problem 4 - first initial point. From the same Table 1 we can also see that the performance of the Alg.2 depends on the initialization of the filter and so far we could not conclude which initialization is better. In all cases except Problem 1 the global minimizers are reached. For Problem 1 with both initial points, Alg. 1 does not converge to the global minimizer while Alg.2(I) - the first initial point and Alg.2 (II) - the first and second initial points reach the global minimizer.

			Alg.1		Alg.2(I)			Alg.2(II)		
prb	n	x_0	iter	gcalc	iter	gcalc	ftr	iter	gcalc	ftr
1	2	(-1, 1)	45	173	33	119	1	41	164	2
		(5, 5)	46	177	107	393	3	47	188	2
2	3	(0, 0, 0)	27	103	32	127	1	35	133	3
		(-1, 1, 1)	45	184	32	119	2	32	119	2
3	4	(0.5, 0.5, 0.5, 0.5)	659	3126	335	1468	3	361	1640	2
		(1.5, 0.5, 1.5, 0.5)	619	2661	852	3754	5	2123	10992	3
4	2	(1, 1)	104	738	98	650	2	52	299	2
		(5, 5)	145	1010	336	2727	3	336	2727	3
4	4	(1, 1, 1, 1)	104	738	104	738	0	42	245	1
		(5, 5, 5, 5)	188	1378	188	1378	0	22	109	1
5	2	(0.5, 0.5)	239	988	511	2091	5	511	2091	5
		(1.2, 1.2)	242	1141	178	755	3	253	1129	4
5	4	(0.5, 0.5, 0.5, 0.5)	266	1065	1712	7713	2	1712	7713	2
		(1.2, 1.2, 1.2, 1.2)	288	1211	265	1107	3	285	1246	3

Table 1. Comparison of GNbBFGS methods

prb	n	x_0		Alg.1	Alg.2(I)	Alg.2(II)
1	2	(-1, 1)	$x(1)$	1.257618339	1.546342884	1.067346967
			$x(2)$	0.795151479	1.391176313	0.139230758
		(5, 5)	$x(1)$	1.257619385	1.25761907	1.067346082
			$x(2)$	0.795153222	0.795152509	0.139227653
2	3	(0, 0, 0)	$x(1)$	0.908926356	0.908926369	0.908926367
			$x(2)$	1.085600025	1.085600012	1.085600018
			$x(3)$	0.682147237	0.68214726	0.682147259
		(-1, 1, 1)	$x(1)$	0.908926364	0.908926365	0.908926365
			$x(2)$	1.085600016	1.085600015	1.085600015
			$x(3)$	0.682147253	0.682147254	0.682147254
3	4	(0.5, 0.5, 0.5, 0.5)	$x(1)$	1.000082461	1.000051394	0.999999082
			$x(2)$	1.000161982	1.000102612	0.999998153
			$x(3)$	0.999929022	0.999949906	1.00000093
			$x(4)$	0.999855112	0.999899011	1.000001867
		(1.5, 0.5, 1.5, 0.5)	$x(1)$	1.000000103	0.999998594	1.00000006
			$x(2)$	1.000000204	0.999997189	1.00000012
			$x(3)$	0.999999899	1.000001296	0.999999941
			$x(4)$	0.999999796	1.000002592	0.999999881
4	2	(1, 1)	$x(1)$	3.40169E-09	9.37398E-09	-2.33E-12
			$x(2)$	3.40169E-09	9.37398E-09	-2.563E-12
		(5, 5)	$x(1)$	-1.9795E-11	-5.99218E-09	-5.99218E-09
			$x(2)$	-1.5949E-11	2.15731E-07	2.15731E-07
4	4	(1, 1, 1, 1)	$x(1)$	3.40169E-09	3.40169E-09	1.3359E-07
			$x(2)$	3.40169E-09	3.40169E-09	1.3359E-07
			$x(3)$	3.40169E-09	3.40169E-09	1.3359E-07
			$x(4)$	3.40169E-09	3.40169E-09	1.3359E-07
		(5, 5, 5, 5)	$x(1)$	2.47E-13	2.47E-13	-4.42473E-07
			$x(2)$	2.47E-13	2.47E-13	-4.42473E-07
			$x(3)$	2.47E-13	2.47E-13	-4.42473E-07
			$x(4)$	2.47E-13	2.47E-13	-4.42473E-07
5	2	(0.5, 0.5)	$x(1)$	0.9999993	0.999996589	0.999996589
			$x(2)$	0.999998595	0.999993156	0.999993156
		(1.2, 1.2)	$x(1)$	1.000001017	0.999997873	0.999999952
			$x(2)$	1.000002031	0.999995733	0.999999903
5	4	(0.5, 0.5, 0.5, 0.5)	$x(1)$	1.000000433	1.000000062	1.000000062
			$x(2)$	1.000000863	1.000000124	1.000000124
			$x(3)$	1.00000172	1.000000249	1.000000249
			$x(4)$	1.000003444	1.000000498	1.000000498
		(1.2, 1.2, 1.2, 1.2)	$x(1)$	0.999999591	1.000000078	1.000000008
			$x(2)$	0.999999187	1.000000154	1.000000007
			$x(3)$	0.99999837	1.000000306	1.000000003
			$x(4)$	0.999996727	1.000000614	1.000000003

Table 2. Final points reached by the tested algorithms

References

[1] M. M. Ali, Ch. Khompatraporn, Z. B. Zabinsky, *A numerical evaluation of several stochastic algorithams on selected continuous global optimization test problems,* Journal of Global Optimization, (2005) 31, pp.635-672

[2] C. Brezinski, *A clasification of quasi-Newton methods,* Numerical Algotithms, (2003) 33, pp.123-135

[3] C. G. Broyden, *A class of methods for solving nonlinear simultaneous equations,* Mathematics of Computation, (1965) 19, pp.577-593

[4] R. H. Byrd, J. Nocedal, *A tool for the analysis of quasi-Newton methods with application to unconstrained minimization,* SIAM J. Numer. Anal., Vol. 26, No. 3 (1989), pp.727-739

[5] C. M. Chin, *A global convergence theory of a filter line search method for nonlinear programming,* Numerical Optimization Report, Department of Statistics, University of Oxford, England, UK, (2002)
`http://www.optimization-online.org/DB_HTML/2002/12/583.html`

[6] Y. Deng, Zh. Liu, *Two derivative-free algorithms for nonlinear equations,* Optimization Methods and Software, Vol. 23, No. 3 (2008), pp.395-410

[7] J. E. Dennis Jr., J. J. Moré, *Quasi-Newton methods, motivation and theory,* SIAM Review 19 (1977), pp.46-89

[8] J. E. Dennis Jr., R. B. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations,* Prentice-Hall, Englewood Cliffs (1983)

[9] R. Fletcher, S. Leyffer, *Nonlinear programming without a penalty function,* Math. Program., 91 (2002), pp.239-270

[10] R. Fletcher, S. Leyffer, Ph. Toint, *A brief history of filter methods,* SIAG/OPT Views-and-News, A Forum for the SIAM Activity Group on Optimization, Vol. 18, No. 1 (2007), pp.2-12

[11] C. C. Gonzaga, E. Karas, M. Vanti, *A globally convergent filter method for nonlinear programming ,* SIAM J. Optim., Vol. 14, No. 3 (2003), pp.646-669

[12] N. I. M. Gould, S. Leyffer, Ph. L. Toint, *A multidimensional filter algorithm for nonlinear equations and nonlinear least-squares,* SIAM J. Optim., Vol. 15, No. 1 (2004), pp.17-38

[13] N. I. M. Gould, Ph. L. Toint, C. Sainvitu, *A filter-trust-region method for unconstrained optimization,* SIAM J. Optim., Vol. 16, (2006), pp.341-357

[14] E. W. Karas, A. P. Oening, A. A. Ribeiro, *Global convergence of slanting filter methods for nonlinear programming,* Applied Mathematics and Computation, Vol. 200, (2008), pp.486-500

[15] D. Li, M. Fukushima, *A globally and superlinearly convergent Gauss-Newton-based BFGS method for symmetric nonlinear equations,* SIAM J. Numer. Anal., Vol. 37, No. 1 (1999), pp.152-172

[16] J. M. Martinez, *Practical quasi-Newton methods for solving nonlinear systems,* Journal of Comp. and Appl. Math., Vol. 124 (2000), pp.97-122

[17] J. Nocedal, S. J. Wright, *Numerical optimization,* Springer-Verlag, New York (1999)

[18] A. Wächter, L. T. Biegler, *Line search filter methods for nonlinear programming: motivation and global convergence,* SIAM J. Optim., Vol. 16, No. 1 (2005), pp.1-31