An Improved Direction of Gradient-type Method for Large Scale Unconstrained Optimization

Mahboubeh, Farid; Malekmohammadi, Najmeh

Published in:
Journal of Advanced Mathematics and Applications

Publication date:
2019

Document Version
Peer reviewed version

Link back to DTU Orbit

Citation (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.
An Improved Direction of Gradient-type Method for Large Scale Unconstrained Optimization

Mahboubeh Farid¹, Najmeh Malekmohammadi²

¹Department of Applied Mathematics and Computer Science, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark
²Department of Mathematics, Faculty of Science, South Tehran Branch, Islamic Azad University, Tehran, Iran
Email: mfarid7@gmail.com, n.malekmohammadi@gmail.com

Abstract

In this paper, a new modification of diagonal-gradient-type method for large scale unconstrained optimization is proposed. We utilize information from the proceeding iteration and consider some corrections for the difference of iterates to improve the current Hessian approximation in diagonal form. Also, the global convergence, under mild conditions is established. Finally, we report some numerical results to show the efficiency of our proposed method.

Key words: Gradient-type methods, diagonal updating, quasi-Newton method, large-scale unconstrained optimization

1 Introduction

Problem of unconstrained optimization is to minimize a nonlinear function \( f(x) \), that is
\[
\min f(x), \quad x \in \mathbb{R}^n
\]
where \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) is twice continuously differentiable. The line search algorithm for problem (1) is iterative in which at the \( k \)th iteration a current approximate solution \( x_k \) is available and the update scheme has the form
\[
x_{k+1} = x_k + \alpha_k d_k \quad k = 0, 1, 2, \ldots,
\]

*Corresponding author.
where \( d_k \) is a descent direction at \( x_k \) and \( \alpha_k \) is stepsize, that is defined by some line search procedures \([3], [5]\). Throughout the paper, \( \|\cdot\| \) denotes the Euclidean norm on \( \mathbb{R}^n \). For convenience, we denote \( f(x_k) \) by \( f_k \) and the gradient of \( f \) at \( x_k \), \( \nabla f(x_k) \) by \( g_k \), respectively. In the class of secant (quasi-Newton) method, \( d_k = -B_k^{-1}g_k \) is used, where \( B_k \) is an approximation to the Hessian matrix at \( x_k \), \( \nabla^2 f(x_k) \). Usually, \( B_k \) is updated in terms of the vector pair \( s_k = x_{k+1} - x_k \) and \( y_k = g_{k+1} - g_k \) to a new approximation \( B_{k+1} \) such that \( B_{k+1}s_k = y_k \) holds \([10]\). Recently, diagonal gradient type methods are proposed for solving large scale problems \([7]\) and \([8]\). They considered letting \( B_k \) as a diagonal matrix to approximate the Hessian at \( x_k \). In general their approach is to require that the approximation \( B_{k+1} \) to satisfy the weak secant equation as follow,

\[
s_k^T B_{k+1} s_k = s_k^T y_k. \tag{3}
\]

The updated \( B_{k+1} \) can be obtained by using a diagonal updating formula, like that in \([8]\) which gives

\[
B_{k+1} = B_k + \xi_k E_k \tag{4}
\]

where \( E_k = \text{diag}((s_k^{(1)})^2, (s_k^{(2)})^2, ..., (s_k^{(n)})^2) \), \( \xi_k = (s_k^T y_k - s_k^T B_k s_k)/\text{tr}(E_k^2) \). The advantage of diagonal gradient type method is that, since the storage of \( B_{k+1} \) is \( O(n) \), which is the same as vector, then they can avoid the computation and storage of matrices associated with the Newton type methods and thus suitable for use in solving large scale optimization. The main objective of this paper is to construct an improved version of the Hessian approximation within the diagonal updating scheme by exploiting information on the difference of iterates, and incorporate them to derive “correction” for the difference of iterates. Hence, we will be able to use the corrected difference of iterates to obtain better updates. This paper is organized as follows. In the next section, we construct our new proposed method. Global convergence of the new algorithm is established in section 3. In section 4, we report some experiments on unconstrained optimization problems and finally, the paper is concluded with some remarks in the last section.

## 2 Diagonal gradient method with corrected direction

In this section, we consider the following updating scheme

\[
x_{k+1} = x_k - \alpha_k B_k^{-1} g_k, \tag{5}
\]

where \( \alpha_k \) is a step size and \( B_k \) is an approximation to the Hessian. The fundamental aim in this section is to modify the weak secant equation (4) by means of the
difference iterates such that it carries more information to improve the approximation for Hessian in diagonal form. Recently, Viček and Lukšan [11] proposed the new correction parameter for $s_k$ and $y_k$ in secant equation and applied it for BFGS method. We are intended to apply their approach for the weak secant equation and derive a better approximation for Hessian in diagonal form. Consider the corrected quantities $\bar{s}_k$, $\bar{y}_k$, $\bar{b}_k$, for all $k \geq 0$ where $\bar{s}_0 = s_0$, $\bar{y}_0 = y_0$, for $k > 0$ $\bar{b}_0 = b_0$ and

$$
\bar{s}_k = s_k - \alpha_k \bar{s}_{k-1},
$$
(6)

$$
\bar{y}_k = y_k - \beta_k \bar{y}_{k-1},
$$
(7)

$$
\bar{b}_k = \bar{s}_k^T \bar{y}_k,
$$
(8)

with $\alpha_k, \beta_k \in \mathbb{R}$ such that $\bar{b}_k > 0$. A possible choice for $\alpha_k$ and $\beta_k$ are given by Viček and Lukšan [11], where they are chosen such that

$$
\alpha_k = \frac{s_k^T \bar{y}_{k-1}}{b_{k-1}},
$$
(9)

and

$$
\beta_k = \frac{s_k^T \bar{y}_{k-1}}{b_{k-1}}.
$$
(10)

The new modification for a weak secant equation will be derived as follow:

$$(s_k - \alpha_k s_{k-1})^T B_{k+1} (s_k - \alpha_k s_{k-1}) = (s_k - \alpha_k s_{k-1})^T (y_k - \beta_k y_{k-1})
$$
(11)

or in the simplified form,

$$
\bar{s}_k B_{k+1} \bar{s}_k = \bar{s}_k^T \bar{y}_k.
$$
(12)

The vector pair $\bar{s}_k$ and $\bar{y}_k$ will be obtained by (6)and (7) and thus the optimal updating formula for $B_{k+1}$ using the approach of [8] can be given by

$$
B_{k+1} = B_k + \tilde{\xi}_k \tilde{E}_k
$$
(13)

where

$$
\tilde{\xi}_k = \frac{(\bar{b}_k - \bar{s}_k^T B_k \bar{s}_k)}{tr(\bar{E}_k^2)}.
$$
(14)

where $\tilde{E}_k = diag((\bar{s}_k^{(1)})^2, (\bar{s}_k^{(2)})^2, ..., (\bar{s}_k^{(m)})^2)$. The major difficulty of the updating (14) formula is that the updated diagonal matrix $B_{k+1}$ may not be positive definite even if $B_k$ does. Non positive definite will happen if $\tilde{\xi}_k < 0$, therefore we can define a scaling parameter to be incorporated into the updating scheme so that this drawback can be overcome:

$$
\gamma_k = \min\left(\frac{\bar{b}_k}{\bar{s}_k^T B_{k+1} \bar{s}_k}, 1\right)
$$
(15)
Accordingly, we will have

$$\hat{\xi}_k = \frac{(\bar{b}_k - \bar{\gamma}_k \bar{s}_k^T B_k \bar{s}_k)}{\text{tr}(\bar{E}_k^2)}.$$  
(16)

and therefore, $B_{k+1}$ will be updated in each step, by the following formula

$$B_{k+1} = \bar{\gamma}_k B_k + \hat{\xi}_k \bar{E}_k.$$  
(17)

Clearly, the scaling (15) can keep positive definiteness of $B_{k+1}$ in each step because under the scaling, $\hat{\xi}_k$ becomes

$$\hat{\xi}_k = \begin{cases} 0, & \text{if } \bar{b}_k \leq \bar{s}_k^T B_k \bar{s}_k \leq \bar{\gamma}_k \bar{s}_k^T B_k \bar{s}_k \text{ /tr}(\bar{E}_k^2), \\ \frac{(\bar{b}_k - \bar{s}_k^T B_k \bar{s}_k)}{\text{tr}(\bar{E}_k^2)}, & \text{otherwise.} \end{cases}$$

Lastly, based on the above discussion, we can present the following algorithm.

**DVMD Algorithm**

**Step 0.** Choose an initial point $x_0 \in \mathbb{R}^n$, $B_0 = I$, and $0 < \sigma_1 < \sigma_2 < 1$. Set $k := 0$.

**Step 1.** Compute $g_k$. If $\|g_k\| \leq \epsilon$, then terminate; Otherwise go to the next step.

**Step 2.** If $k = 0$, set $x_1 = x_0 - \frac{\omega}{\|g_0\|}$ and let $\bar{s}_0 = s_0$, $\bar{y}_0 = y_0$, $\bar{b}_0 = b_0$.

If $k \geq 1$, compute $\tilde{b}_k > 0$, $\alpha_k$ and $\beta_k$ from (8),(9) and (10), respectively. Set $\bar{s}_k = s_k - \alpha_k s_{k-1}$ and $\bar{y}_k = y_k - \beta_k y_{k-1}$.

Else if $\tilde{s}_k^T \bar{y}_k \leq \tilde{\epsilon} \|\bar{s}_k\|_2 \|\bar{y}_k\|_2$, (to safeguard very small $\tilde{s}_k^T \bar{y}_k$, here we use $\tilde{\epsilon} = 10^{-4}$) set $\bar{s}_k = s_k$ and $\bar{y}_k = y_k$.

**Step 3.** Compute $d_k = -B_k^{-1} g_k$ and determine a stepsize $\alpha_k > 0$ using the Wolfe line search:

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \sigma_1 \alpha_k g_k^T d_k$$  
(18)
$$g(x_k + \alpha_k d_k)^T d_k \geq \sigma_2 g_k^T d_k$$  
(19)

**Step 4.** Let $x_{k+1} = x_k + \alpha_k d_k$ and update $B_{k+1}$ by (17).

**Step 5.** Set $k := k + 1$ and return to Step 1.

### 3 Convergence Analysis

This section studies the convergence of DVMD algorithm, when applied to the minimization of a convex function. We begin by giving the following well-known result of Byrd et al. [4] for the steps generated by the Armijo-type line search algorithm. Here and thereafter, $\| \cdot \|$ denotes the Euclidean norm.
**Theorem 3.1** Assume that $f$ is a strictly convex function. Suppose the line search algorithm is employed in a way that for any $d_k$ with $d_k^T g_k < 0$, the stepsize, $\alpha_k$ satisfies the Wolfe conditions (18) and (19). Then there exist positive constants $\rho_1$ and $\rho_2$ such that either
\[ f(x_k + \alpha_k d_k) - f(x_k) \leq -\rho_1 \frac{(d_k^T g_k)^2}{\|d_k\|^2} \] (20)
or
\[ f(x_k + \alpha_k d_k) - f(x_k) \leq -\rho_2 d_k^T g_k \] (21)
is satisfied.

The following result can be obtained using Theorem 3.1.

**Theorem 3.2** Assume that $f$ is a strictly convex function. Suppose that the line search algorithm in Theorem 3.1 is employed with $d_k$ chosen to obey the following conditions: There exist positive constants $c_1$ and $c_2$ such that
\[ -g_k^T d_k \geq c_1 \|g_k\|^2, \quad \text{and} \quad \|d_k\| \leq c_2 \|g_k\|, \] (22)
for all sufficiently large $k$. Then the iterates $x_k$ generated by the line search algorithm have the property that
\[ \liminf_{k \to \infty} \|g_k\| = 0. \] (23)

**Proof.** By (22), we have that, either (20) or (21) will become
\[ f(x_k + \alpha_k d_k) - f(x_k) \leq -c \|g_k\|^2, \] (24)
for some positive constant. Since $f$ is strictly convex and it is also bounded below, then (18) implies that $f(x_k + \alpha_k d_k) - f(x_k) \to 0$ as $k \to \infty$. This also implies that $\|g_k\| \to 0$ as $k \to \infty$ or at least
\[ \liminf_{k \to \infty} \|g_k\| = 0. \] (25)

We can now apply Theorem 3.1 and 3.2 to establish the convergence of DVMD Algorithm when applied to the minimization of a convex function. This can be done by showing that the sequence $\{|B_k|\}$ generated by (17) is bounded both above and below, for all finite $k$ so that its associated search direction satisfies condition (22). The following theorem gives the boundedness of $\{|B_k|\}$:
**Theorem 3.3** Assume that \( f \) is strictly convex function where there exists positive constants \( m \) and \( M \) such that

\[
m\|z\|^2 \leq z^T \nabla^2 f(x) z \leq M\|z\|^2,
\]

for all \( z \in \mathbb{R}^n \). Let \( \{\|B_k\|\} \) be a sequence generated by the DVMD method. Then \( \|B_k\| \) is bounded both above and below for all finite \( k \), by some positive constants.

**Proof.** Let \( B_k^{(i)} \) be the \( i \)-th element of \( B_k \). Suppose \( B_0 \) is chosen such that \( \omega_1 \leq B_0^{(i)} \leq \omega_2, \quad \forall i = 1, \ldots, n \) where \( \omega_1 \leq \omega_2 \) are some positive constants. If \( \bar{b}_k \leq \bar{s}_k^T B_k \bar{s}_k \) (or \( \xi_0 = 0 \)), we have

\[
B_1 = \gamma_0 B_0.
\]

Then by the definition of \( \gamma_0 \) and (26), one can obtain

\[
m\omega_1/\omega_2 \leq B_1^{(i)} \leq \omega_2.
\]

On the other hand, if \( \bar{b}_k > \bar{s}_k^T B_k \bar{s}_k \), then

\[
B_1^{(i)} = B_0^{(i)} + \frac{\bar{s}_0^T \bar{y}_0 - \bar{s}_0^T B_0 \bar{s}_0}{\text{tr}(E_0^2)} (\bar{s}_0^{(i)})^2,
\]

where \( \bar{s}_0^{(i)} \) is the \( i \)-th component of \( \bar{s}_0 \). Denoting \( (\bar{s}_0^{(M)}) \) be the largest component (in magnitude) of \( \bar{s}_0 \), i.e. \( (\bar{s}_0^{(i)})^2 \leq (\bar{s}_0^{(M)})^2 \); \( \forall i \), and by the safeguarding strategy in Step 2 of the algorithm, i.e. \( \bar{s}_0^T \bar{y}_0 \leq \bar{\varepsilon}_s \|\bar{s}_0\| \|\bar{y}_0\| \), (17) becomes

\[
\omega_1 \leq B_0^{(i)} \leq B_1^{(i)} \leq \omega_2 + \frac{\|\bar{s}_0\|\|\bar{y}_0\| - \omega_1\|\bar{s}_0\|^2}{\text{tr}(E_0^2)} (\bar{s}_0^{(M)})^2.
\]

Since

\[
\|\bar{y}_k\|^2 = \left\| \left( \int_0^1 \nabla^2 f(x_k + \tau \bar{s}_k) d\tau \right) \bar{s}_k \right\|^2 \leq M^2 \|\bar{s}_k\|^2, \quad \forall k,
\]

then together with the fact that \( \|\bar{s}_0\|^2 \leq n(\bar{s}_0^{(M)})^2 \) and \( \text{tr}(E_0^2) = \sum_{i=1}^n (\bar{s}_0^{(i)})^4 \geq (\bar{s}_0^{(M)})^4 \), it also implies that

\[
\frac{(\bar{\varepsilon}_s - \omega_1)\|\bar{s}_0\|^2}{\text{tr}(E_0^2)} (\bar{s}_0^{(M)})^2 \leq \frac{n(\bar{\varepsilon}_s - \omega_1)}{\text{tr}(E_0^2)} (\bar{s}_0^{(M)})^4 \leq n(\bar{\varepsilon}_s - \omega_1).
\]

(In case when the safeguarding condition in Step 2 is violated, we can set \( \bar{s}_0^T \bar{y}_0 = \bar{s}_0^T y_0 \leq M\|s_0\| \) and establish the upper bound for \( B_1^{(i)} \) as \( n(M - \omega_1) \).) Hence, \( B_1^{(i)} \) is bounded both above and below, namely

\[
\omega_1 \leq B_1^{(i)} \leq \omega_2 + \leq n(\bar{\varepsilon}_s - \omega_1).
\]

Also since in both cases, the upper and lower bound for \( B_1^{(i)} \) is, respectively independent of \( k \), we can proceed by using induction to show that \( B_k^{(i)} \) is bounded, for all finite \( k \).
4 Numerical Results

This section is devoted to test the implementation of the our proposed method for solving large scale problems. We compare the performance of the DVMD method with the SMD method which is the scaled standard one-step method in [?]. All the experiments in this paper are implemented on a PC with a Core Duo processor and using MATLAB 7. In all the tests, the values of $\sigma_1 = 0.1$ and $\sigma_2 = 0.8$ are used in conditions (18) and (19) to determine the step length $\alpha_k$. The value $\epsilon = 10^{-4}$ is used in termination test. We also force the routine to stop if the number of iteration exceed 1000. Some classical test functions with standard starting points are selected to test the two diagonal updating methods (see [1],[6]). The dimensions of these problems are varying from $n = 100$ to $10^5$. We have used the performance profiles proposed by Dolan and More’ [6] to display the performance of each algorithm, in terms of function and gradient evaluations. The use of this profile provides a wealth of information such as solver efficiency, robustness and probability of success in compact form and eliminate the influence of a small number of problems on the benchmarking process and the sensitivity of results associated with the ranking of solvers [6]. The performance profile plots the fraction $P$ of problems for which any given method is within a factor $\tau$ of the best solver. The horizontal axis of the figure gives the percentage of the test problems for which a method is the fastest (efficiency), while the vertical axis gives the percentage of the test problems that were successfully solved by each method (robustness)

Table 1: Test problem and its dimension

<table>
<thead>
<tr>
<th>Problem</th>
<th>Dimension</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extended Trigonometric, Penalty 1, Penalty 2, Full Hessian FH1,</td>
<td>100,1000,5000,10000,100000</td>
<td>More’ et al. [9]</td>
</tr>
<tr>
<td>Extended Beale, Broyden Tridiagonal, Quadratic Diagonal Perturbed,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Perturbed Quadratic, Quadratic QF1, Diagonal 1, Diagonal 2, Hager,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diagonal 3, Generalized Tridiagonal 2, Almost perturbed Quadratic,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tridiagonal perturbed quadratic, Full Hessian FH2,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Raydan 1, EG2, Extended White and holst, Extended Rosenbrock,</td>
<td></td>
<td>Andrei [1]</td>
</tr>
<tr>
<td>Quadratic QF2, Diagonal 4, Diagonal 5, Generalized Tridiagonal 1,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Generalized Rosenbrock, Generalized PSC1, Extended Himmelblau,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extended Three Exponential Terms, Extended Block Diagonal BD1,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extended PSC1, Raydan 2, Extended Tridiagonal 2, Extended Powell</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extended Freudenstein and Roth.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 1 presents the performance profiles of DVMD and the SMD, relative to the number of iteration. Clearly, DVMD exhibits the best overall performance since it illustrates the best probability of being the optimal solver, outperforming SMD,
relative to the iteration metric. Moreover, the performance profile for iteration shows that DVMD solves more than 70 percent of the test problems with the least number of iteration while SMD solve about 50 percent of the test problems. This figure indicates that DVMD seems to be the best by comparison with SMD and need less iteration. This is not surprising when we consider that the modified diagonal updating always get a better approximation of the Hessian matrix at each iteration. Fig. 2 shows the implementation of the DVMD and the SMD method using the function evaluation in each iteration as a measure. It also shows that the DVMD method needs less function evaluation than the SMD. We observe that the iterative form of DVMD is very simple and requires low storage and at each iteration, our proposed method does not require more storage than the SMD. In fact; this new method just needs $O(n)$ storage. Moreover, a higher order accuracy in approximating the Hessian matrix of the objective function makes DVMD need less iterations, less function evaluations.

![Figure 1: Performance profile based on Iteration for all problems.](image)

## 5 Conclusion

We have presented and analyzed a algorithm for unconstrained optimization. we proposed some modifications of weak secant equation based on correction of difference vector which provide better approximation of Hessian. The new method can guarantee a descent at each iteration. We are able to show that the proposed method is globally convergent under mild conditions. Numerical results show that
the proposed method is suitable to solve large-scale unconstrained optimization problems and more stable than standard diagonal updating method in practical computation.

References


