Applied Numerical Mathematics 91 (2015) 75–88

Contents lists available at ScienceDirect

Applied Numerical Mathematics

www.elsevier.com/locate/apnum

Nonmonotone adaptive trust region method with line search based on new diagonal updating ☆

Qunyan Zhou [a,*], Dan Hang [b]

[a] School of Mathematics and Physics, Jiangsu University of Technology, Changzhou 213001, China
[b] Department of Basic Courses, Air Force Logistics College, Xuzhou 221000, China

ARTICLE INFO

Article history:
Received 5 August 2013
Received in revised form 16 December 2014
Accepted 21 December 2014
Available online 7 January 2015

Keywords:
Diagonal updating
Weak secant equation
Nonmonotone technique
Adaptive trust region method
Global convergence

ABSTRACT

In this paper, a new nonmonotone adaptive trust region method with line search for solving unconstrained nonlinear optimization problems is introduced. The computation of the Hessian approximation is based on the usage of the weak secant equation by a diagonal definite matrix. Under some reasonable conditions, the global convergence of the proposed algorithm is established. The numerical results show the new method is effective and attractive for large scale optimization problems.

1. Introduction

Consider the following large scale unconstrained optimization problem

$$\min_{x\in R^n} f(x), \tag{1.1}$$

where $f(x): R^n \to R$ is continuously differentiable.

Line search and trust region methods are two prominent classes of iterative methods to solve the unconstrained optimization problem (1.1). For a given iteration point x_k, the line search method has the form defined by the equation $x_{k+1} = x_k + \alpha_k d_k$ to derive a new point, where d_k is a descent direction of $f(x)$ at x_k and α_k is the stepsize. In the Armijo-type line search method, α_k satisfies

$$f(x_k + \alpha_k d_k) \le f_k + \gamma \alpha_k g_k^T d_k, \tag{1.2}$$

where $\gamma \in (0, 1)$, $f_k = f(x_k)$, $g_k = \nabla f(x_k)$. On the other hand, the trust region methods usually calculate a trial step d_k by solving the following quadratic subproblem

$$\begin{aligned} \min \quad & q_k(d) = f_k + g_k^T d + \tfrac{1}{2} d^T B_k d, \\ \text{s.t.} \quad & \|d\| \le \Delta_k, \end{aligned} \tag{1.3}$$

☆ This work is supported by National Natural Science Foundation of China (11471145), the Research Program of Natural Science of Universities in Jiangsu Province (13KJB110007) and the Foundation of Jiangsu University of Technology (KYY13012, KYY14010).

* Corresponding author.
E-mail addresses: zhouqunyan@jsut.edu.cn (Q. Zhou), hazel-hang@sohu.com (D. Hang).

http://dx.doi.org/10.1016/j.apnum.2014.12.009
0168-9274/

where $B_k \in R^{n\times n}$ is a symmetric matrix which is the Hessian matrix or its approximation of $f(x)$ at the current point x_k, $\Delta_k > 0$ is called the trust region radius and $\|\cdot\|$ refers to the Euclidean norm. The ratio r_k between the actual reduction in the function value $f_k - f_{k+1}$ and the predicted reduction $q_k(0) - q_k(d_k)$ plays a key role to decide whether the trial step is acceptable or not and how to adjust the trust region radius. When the trial step is not successful, one rejects it, reduces the trust region radius and resolves the subproblem, which can be costly. Nocedal and Yuan [12] proposed a new type of trust region algorithms that take the advantage of combining line search to find an iterative point instead of resolving the trust region subproblem. Besides, the adjustment strategy, in which the trust region radius is updated only by simply enlarging or reducing the initial trust region radius at a constant rate, does not make full use of the information at the current iterate point, such as the first-order and second-order derivatives. Hence, many authors have studied the self-adaptive trust region method [4,22]. In [14], a new efficient self-adaptive adjustment strategy for updating the trust region radius was proposed. That is, given $0 \le \mu_1 < \mu_2 < 1$, $0 < c_2 < 1 < c_1$, set

$$\Delta_{k+1} = \theta_{k+1}\|g_{k+1}\|\left\|B_{k+1}^{-1}\right\|, \tag{1.4}$$

where

$$\theta_{k+1} = \begin{cases} c_1\theta_k, & \text{if } r_k > \mu_2; \\ c_2\theta_k, & \text{if } r_k < \mu_1; \\ \theta_k, & \text{if } \mu_1 \le r_k \le \mu_2. \end{cases}$$

Grippo, Lampariello and Lucidi [9] found that the methods requiring monotonically decreasing of the objective function values at each iteration may slow the rate of convergence in the presence of a narrow valley. They proposed a nonmonotone line search technique for Newton's method, in which the stepsize α_k satisfies the following inequality

$$f(x_k + \alpha_k d_k) \le f_{l(k)} + \gamma\alpha_k g_k^T d_k, \tag{1.5}$$

where $f_{l(k)} = \max_{0\le j\le m_k}\{f_{k-j}\}, m_0 = 0, 0 \le m_k \le \min\{m_{k-1}+1, M\}$ $(k \ge 1)$, and $M \ge 0$ is an integer. This nonmonotone technique was generalized to the trust region method in [6]. However, it has some disadvantages. For example, it follows from (1.5) that a good function value generated at any iteration may be thrown away due to the maximum, and the numerical results are dependent on the choice of parameter M. In order to overcome these disadvantages, Zhang and Hager [20] proposed another nonmonotone line search. They replaced the maximum function value in (1.5) with an average of function values, that is, their nonmonotone technique requires decreasing of an average of the successive function values. In detail, their method finds the stepsize α_k satisfying the following condition

$$f(x_k + \alpha_k d_k) \le C_k + \gamma\alpha_k g_k^T d_k, \tag{1.6}$$

where

$$C_k = \begin{cases} f_k, & k = 0, \\ \frac{\eta_{k-1}Q_{k-1}C_{k-1}+f_k}{Q_k}, & k \ge 1, \end{cases} \qquad Q_k = \begin{cases} 1, & k = 0, \\ \eta_{k-1}Q_{k-1}+1, & k \ge 1, \end{cases} \tag{1.7}$$

and $\eta_{k-1} \in [\eta_{\min}, \eta_{\max}]$, $\eta_{\min} \in [0, 1)$ and $\eta_{\max} \in [\eta_{\min}, 1)$ are two chosen parameters. Numerical results showed that this nonmonotone technique was superior to (1.5). Then, this nonmonotone technique was also applied to the trust region methods [11,17]. Recently, Gu and Mo [10] found that updating η_k and Q_k at each iteration becomes an encumbrance. To overcome this limitation, Gu and Mo [10] introduced another nonmonotone strategy. They replaced C_k in (1.6) with R_k which is a simple convex combination of the previous R_{k-1} and f_k, i.e.,

$$R_k = \begin{cases} f_k, & k = 1, \\ \eta_k R_{k-1} + (1-\eta_k) f_k, & k \ge 2 \end{cases} \tag{1.8}$$

for $\eta_k \in [\eta_{\min}, \eta_{\max}]$. Numerical experiments in [10] show that this nonmonotone technique is efficient and robust.

In the quasi-Newton algorithm framework, the Hessian approximation matrix B_{k+1} is usually required to satisfy the secant equation

$$B_{k+1}s_k = y_k, \tag{1.9}$$

where $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$. One of the widely used quasi-Newton method to solve general nonlinear minimization is the BFGS method, which uses the following updating

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{s_k^T y_k}. \tag{1.10}$$

On the numerical aspect, this method supersedes most of the optimization methods. However, it needs $O(n^2)$ storage which makes it unsuitable for large scale problems. It is necessary to modify and extend this method to make it suitable