



COPYRIGHT NOTICE

UB ResearchOnline
<http://researchonline.ballarat.edu.au>

This is an Author's Accepted Manuscript of an article published in
TaHERi, S., Mammadov, M., Seifollahi, S. (2013) Globally convergent
algorithms for solving unconstrained optimization problems.
Optimization, available online at:

<http://www.tandfonline.com/10.1080/02331934.2012.745529>

Copyright © Taylor & Francis Group

RESEARCH ARTICLE

Globally Convergent Algorithms for Solving Unconstrained Optimization Problems

Sona Taheri^{a*} & Musa Mammadov^{ab} and Sattar Seifollahi^c

^a*Centre for Informatics and Applied Optimization, School of Science, Information Technology and Engineering, University of Ballarat, VIC 3353, Australia;* ^b*National ICT Australia;* ^c*School of Civil, Environmental and Mining Engineering, University of Adelaide, SA 5005, Australia*

(released March 2012)

New algorithms for solving unconstrained optimization problems are presented based on the idea of combining two types of descent directions: the direction of anti-gradient and either the Newton or Quasi-Newton directions. The use of latter directions allows one to improve the convergence rate. Global and superlinear convergence properties of these algorithms are established. Numerical experiments using some unconstrained test problems are reported. Also the proposed algorithms are compared with some existing similar methods using results of experiments. This comparison demonstrates the efficiency of the proposed combined methods.

Keywords: Unconstrained optimization; Gradient method; Newton method; Quasi-Newton method; Global convergence; Superlinear convergence

AMS Subject Classifications: 90C53; 49M15; 65K05

1. Introduction

Consider the following unconstrained minimization problem

$$\min f(x) \text{ subject to } x \in \mathbb{R}^n \quad (1)$$

where the function f is twice continuously differentiable. Let $g(x) = \nabla f(x)$ and $H(x) = \nabla^2 f(x)$ be the gradient and the Hessian matrix of the function f , respectively.

Numerical methods have been developed extensively for solving the minimization problem (1). The gradient method is one of the simplest and commonly used methods. Although this method is globally convergent, it suffers from the slow convergence rate as a stationary point is approached. In order to improve the convergence rate, one can use the Newton method. This method is one of the most popular methods due to its attractive quadratic convergence, but it depends on the initial point and sometimes the computation of the inverse of the Hessian could be time consuming [18]. A number of different modified Newton methods have been introduced to improve the performance of the Newton method [1, 8–10, 12]. However, the global convergence of these methods are not always guaranteed.

In order to avoid these difficulties, one way is the use of a combination of different local optimization methods. In recent years, there has been a growing interest

*Corresponding author. Email: sonataheri@students.ballarat.edu.au

in applying such combined methods. De Luca et al. [5] considered a combination of the gradient and the Newton methods for the solution of nonlinear complementarity problems. The work of Malik HJ et al. [13], employs a hybrid descent direction strategy which uses a convex combination of the anti-gradient and the Quasi-Newton as a search direction. Buckley [2, 3] proposed a strategy of using the Conjugate gradient search direction for the most iterations and periodically using the Quasi-Newton direction to improve the convergence. Wang et al. [19] proposed a revised Conjugate gradient projection method, that is, a combination of the Conjugate gradient projection and the Quasi-Newton methods for nonlinear inequality constrained optimization problems. Shi [16] introduced a method based on the combination of the gradient and Newton methods for solving a system of nonlinear equations. In [4], he proposed a combination of the modified Quasi-Newton and the gradient method to find a solution for systems of linear equations. Furthermore, Shi developed methods based on the combinations of the gradient method with Newton and Quasi-Newton methods for solving unconstrained optimization problems [17]. Recently, the idea of combining the gradient method with the Newton and the Quasi-Newton methods has been developed in [7, 11, 21]. These combinations are also applied for minimizing the cost function during the training of Neural Networks [7]. More recently Yang [20] applied the Newton-Conjugate gradient method for solitary wave computations.

In this paper, we propose new algorithms based on the idea of combining the anti-gradient direction with either the Newton direction or the Quasi-Newton direction for solving the problem (1). We call the algorithm involving combination of the gradient and Newton methods as Algorithm CGN, and the combination of the gradient and Quasi-Newton methods as Algorithm CGQN. These algorithms are different from the existing combination algorithms [2–5, 7, 11, 13, 16, 17, 19–21]. We introduce a special parameter which allows us to control contribution from each component method. We also define two different combinations. The first one is a novel combination in which the step length, α_k , is determined only along the anti-gradient direction. The second one is similar to those developed in [4, 16, 17]. Under some assumptions we prove that the proposed methods are globally convergent and they have superlinear convergence rate.

The rest of the paper is organized as follows. In the next section, we present a general scheme of the descent methods and some theorems which are used to establish the convergence of the proposed methods. In Section 3, we describe the proposed algorithms in details. The global and superlinear convergence properties of our algorithms are proved in Sections 4 and 5, which is followed by some numerical experiments in Section 6, demonstrating the efficiency of the proposed algorithms. Finally, some concluding remarks are made in Section 7.

2. Preliminaries

Consider the problem (1) and denote by $g_k = \nabla f(x_k)$, the gradient of the function f at a point x_k . A general descent method for solving Problem (1) proceeds as follows:

Algorithm 1: A Descent Method

Initialization. Select a starting point $x_0 \in \mathbb{R}^n$, and a tolerance $\varepsilon > 0$, set $k:=0$.

Step 1. If $\|g_k\| < \varepsilon$, then stop.

Step 2. Compute a descent direction d_k at x_k satisfying

$$g_k^T d_k < 0. \quad (2)$$

Step 3. Determine an appropriate step length $\alpha_k > 0$.

Step 4. Set $x_{k+1} := x_k + \alpha_k d_k$, $k := k + 1$ and go to Step 1.

Depending on the choice of d_k and α_k , where d_k is a descent direction and α_k is a step length, different descent direction methods have been developed. There are two alternatives for finding α_k , namely using the exact and inexact line search. In practical implementations, the finding an exact optimal step length is, in general, difficult or expensive [18], therefore, the inexact line search with less computational load is highly popular. There are some inexact line search techniques such as Armijo, Goldstein and Wolfe-Powell rules. Given descent direction d_k , the Wolfe-Powell rule suggests the following relations to find the step length $\alpha_k > 0$ [18]

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \rho \alpha_k g_k^T d_k, \quad (3)$$

$$g_{k+1}^T d_k \geq \sigma g_k^T d_k, \quad (4)$$

where $\rho \in (0, 1/2)$ and $\sigma \in (\rho, 1)$.

Let us consider the Wolfe-Powell conditions (3) and (4) to determine α_k in the descent direction algorithm. The global convergence of the general descent direction algorithm is given by the following theorem [18].

Theorem 2.1: *Let α_k in the descent direction algorithm be defined by (3) and (4). Let also d_k satisfy*

$$\cos(\theta_k) \geq \delta \quad (5)$$

for some $\delta > 0$ and for all k , where θ_k is the angle between d_k and $-g_k$. If $g(x)$ exists and is uniformly continuous on the level set $\{x \in \mathbb{R}^n \mid f(x) \leq f(x_0)\}$, then either $g_k = 0$ for some k , or $f_k \rightarrow -\infty$, or $g_k \rightarrow 0$.

One of the simplest and the most fundamental minimization methods satisfying Theorem 2.1 is the gradient method, in which $d_k = -g_k$, for all k . Although this method is globally convergent and usually works well in some early steps, as a stationary point is approached, it may descend very slowly.

In order to improve the convergence rate, one can use the Newton method. At the k -th iteration, the classical Newton direction d_k is the solution to the following system:

$$H_k d_k = -g_k, \quad (6)$$

where H_k is the Hessian matrix at x_k . In general, the Newton method is not globally convergent. Moreover, this method requires the computation of the inverse of the Hessian in order to find descent directions which can be time consuming. One technique, for instance, is the Quasi-Newton method which uses approximations with a positive definite matrix. However, these approximations still will not guarantee the global convergence. A common strategy that is recently applied to guarantee the global convergence is the use of methods based on the combination

of different local optimization methods [2–5, 7, 11, 13, 16, 17, 19–21]. Most of these methods are efficient for solving the problem (1) due to their global convergence property and high local convergence rate. We propose new combined methods in the next section.

The following theorems will be used to prove the convergence of the proposed methods. Their proofs can be found in [18].

Theorem 2.2: *Let $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ be continuously differentiable in the open convex set $D \subset \mathbb{R}^n$. Assume that H is Lipschitz continuous in D with a Lipschitz constant $\gamma \geq 0$. Then for any $u, v, x \in D$, we have*

$$\|g(u) - g(v) - H(x)(u - v)\| \leq \gamma \frac{\|u - x\| + \|v - x\|}{2} \|u - v\|. \quad (7)$$

Theorem 2.3: *Let g and H satisfy the conditions of Theorem 2.2. Assume that $H^{-1}(x)$ exists. Then there exist $\varepsilon > 0$ and $\mu > \beta > 0$ such that for all $u, v \in D$, when $\max\{\|u - x\|, \|v - x\|\} \leq \varepsilon$, we have*

$$\beta \|u - v\| \leq \|g(u) - g(v)\| \leq \mu \|u - v\|. \quad (8)$$

3. The Proposed Algorithms

In this section, we introduce our new algorithms, called CGN and CGQN, for solving the unconstrained optimization problem (1). Algorithm CGN is based on the idea of combining anti-gradient and Newton directions. In Algorithm CGQN, we use the Quasi-Newton direction in the combination with the anti-gradient direction.

Throughout the paper $d_{1,k}$ denotes the anti-gradient direction at x_k and $d_{2,k}$ stands for the second direction at x_k to be used in the combination with $d_{1,k}$. The steps of our combined methods are presented in Algorithm 2.

Algorithm 2: Algorithms CGN and CGQN

Initialization. Select a starting point $x_0 \in \mathbb{R}^n$, and a tolerance $\varepsilon > 0$, η and δ be small positive numbers and $\vartheta > 1$, ω and L are two fixed numbers. Set $k := 0$.

Step 1. If $\|g(x_k)\| < \varepsilon$, then stop.

Step 2. Compute the direction $d_{1,k}$ at x_k , $d_{1,k} = -g_k$.

Step 3. Compute a second direction $d_{2,k}$ at x_k . If the direction $d_{2,k}$ at x_k is not computable, then go to Step 5.

Step 4. If $d_{2,k}^T d_{1,k} \geq 0$, go to Step 6.

Step 5. Use rules (3) and (4) to determine a step length $\alpha_k > 0$ along the direction $d_k = d_{1,k}$, set $s_k := \alpha_k d_k$ and go to Step 10.

Step 6. Set $j := 0$, $\eta_0 := \eta$.

Step 7. Compute ξ_k as follows:

$$\xi_k = \begin{cases} \frac{1}{1+\eta_j \|g_0\|} & \text{if } k = 0, \\ \frac{1}{1+\eta_j |f_k - f_{k-1}|} & \text{if } k > 0, \end{cases} \quad (9)$$

and set $d(\xi_k) := (1 - \xi_k)d_{1,k} + \xi_k d_{2,k}$.

Step 8. If $d(\xi_k)^T d_{1,k} < \delta \|d(\xi_k)\| \|d_{1,k}\|$, set $\eta_{j+1} := \vartheta \eta_j$ and $j = j + 1$, and go to Step 7.

Step 9. Compute s_k using one of the following two approaches:

9.1. Use rules (3) and (4) to determine a step length $\alpha_k > 0$ along the direction $d_{1,k}$ and set $\bar{s}_k := \alpha_k(1 - \xi_k)d_{1,k} + \xi_k d_{2,k}$. If $f(x_k + \bar{s}_k) \leq f(x_k) - \omega \|\bar{s}_k\|$ and $\alpha_k \|d_{1,k}\| \leq L \|d_{2,k}\|$, set $s_k := \bar{s}_k$; otherwise set $s_k := \alpha_k d_{1,k}$.

9.2. Use rules (3) and (4) to determine a step length $\alpha_k > 0$ along the direction $d_k = d(\xi_k)$ and set $s_k := \alpha_k d_k$.

Step 10. Set $x_{k+1} := x_k + s_k$, $k := k + 1$ and go to Step 1.

The direction $d_{2,k}$ can be either the Newton direction or the Quasi-Newton direction. In Algorithm CGN, when the Hessian at x_k is singular or $d_{2,k}$ is not computable then we use the anti-gradient direction. Moreover, if $d_{2,k}^T d_{1,k} < 0$ then the Newton direction tends to increase the function value. In this case, again, we take the anti-gradient direction as indicated in Step 5.

In Algorithm CGQN, we use the Quasi-Newton direction as the second direction in the combination with the anti-gradient direction. In the Quasi-Newton method, an approximation B_k is used instead of the Hessian H_k . At the k -th iteration, the Quasi-Newton direction is the solution to the following system:

$$B_k d_k = -g_k, \tag{10}$$

where B_k is a positive definite matrix. There are some well known formulas for updating B_k in the Quasi-Newton method [18]. In this paper, B_k is updated by the BFGS formula as follows:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{s_k^T y_k}, \tag{11}$$

where $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$.

In practical implementations, when $s_k^T y_k = 0$ (or $s_k^T y_k$ is too small), then B_k and consequently $d_{2,k}$ may not be computable and we use only the anti-gradient direction as indicated in Step 5.

In equation (9), parameter ξ_k is in the interval $[0, 1]$ that weights two different directions in the combination. When the slope of the function is slight, the algorithm tends to the second direction, $d_{2,k}$, otherwise it is close to the anti-gradient direction, $d_{1,k}$. More precisely, when the difference between function values is a large number, and consequently ξ_k is close to 0, the gradient method may work better. Also, it is clear from (9) that, near the solution, we can get the optimal point with a high convergence rate.

In Step 7, we consider two different conditions for choosing ξ_k . At the first step, $k = 0$, the value of $|f(x_k) - f(x_{k-1})|$ is not defined, so we will use $\|g_0\|$ instead.

In Step 9, we use two different strategies for the combination. Step 9.1 is a new combination and is different from the existing methods in the literature [4, 16, 17]. In this combination, the step length α_k is determined only along the anti-gradient direction. In this strategy, we define two conditions that make the new direction to be a descent direction. In Step 9.2, the step length α_k is determined along the combination of both directions $d_{1,k}$ and $d_{2,k}$.

4. Global Convergence

The following theorem shows that Algorithms CGN and CGQN are globally convergent.

Theorem 4.1: *Consider using Algorithm 2 to solve the problem (1). Assume that $g(x)$ exists and is uniformly continuous on the level set $\{x \in \mathbb{R}^n \mid f(x) \leq f(x_0)\}$. Then either $g_k = 0$ for some k , or $f_k \rightarrow -\infty$, or $g_k \rightarrow 0$.*

Proof: Let us assume that $g_k \neq 0$ and f_k is bounded below for all k . Clearly in this case $f_k < f_{k-1}$ for all k . We need to show that $g_k \rightarrow 0$.

Denote by θ_k the angle between d_k and $-g_k$. Consider iterations $x_{k+1} = x_k + s_k$. There are two versions to be considered. In the first version s_k is calculated using Steps 5 and 9.2 of Algorithm 2, in the second version s_k is calculated using Steps 5 and 9.1.

Version 1. If s_k is obtained at Step 5, then $d_k = d_{1,k} = -g_k$ and consequently

$$\cos(\theta_k) = \frac{-d_k g_k}{\|d_k\| \|g_k\|} = 1 > \delta. \quad (12)$$

Now suppose s_k is obtained at Step 9.2. Then $d_k = d(\xi_k)$ is chosen as a descent direction and according to Steps 7-8, the number ξ_k can be chosen so that the inequality $d(\xi_k)^T d_{1,k} \geq \delta \|d(\xi_k)\| \|d_{1,k}\|$ holds. Then we have

$$\cos(\theta_k) = \frac{d(\xi_k)^T d_{1,k}}{\|d(\xi_k)\| \|d_{1,k}\|} \geq \delta, \quad (13)$$

Therefore, in this version for all k the inequality $\cos(\theta_k) \geq \delta > 0$ holds and the proof of the theorem follows from Theorem 2.1.

Version 2. Suppose s_k is obtained at Steps 5 and 9.1. In this case we have $x_{k+1} = x_k + s_k$ where s_k is defined by

$$s_k = \alpha_k d_{1,k} \quad (14)$$

or

$$s_k = \alpha_k (1 - \xi_k) d_{1,k} + \xi_k d_{2,k}. \quad (15)$$

Here we note that according to Step 9.1 the step length $\alpha_k > 0$ is determined by the Wolfe-Powell rule along the direction $d_{1,k}$; that is, the following two inequalities are satisfied:

$$f(x_k + \alpha_k d_{1,k}) \leq f(x_k) + \rho \alpha_k g_k^T d_{1,k}, \quad (16)$$

$$g(x_k + \alpha_k d_{1,k})^T d_{1,k} \geq \sigma g_k^T d_{1,k}, \quad (17)$$

where $d_{1,k} = -g_k = -g(x_k)$.

If the number of cases when s_k is defined using (15) is finite, that is, s_k is defined by (14) for all sufficiently large k , then the proof follows from Theorem 2.1 in view

of (12).

Assume that there is a subsequence $k_m \rightarrow \infty$ such that

$$s_{k_m} = \alpha_{k_m}(1 - \xi_{k_m})d_{1,k_m} + \xi_{k_m}d_{2,k_m}. \quad (18)$$

According to Step 9.1, this in particular means that the following two relations hold:

$$f(x_{k_m} + s_{k_m}) \leq f(x_{k_m}) - \omega \|s_{k_m}\|, \quad \text{for all } k_m \quad (19)$$

and

$$\alpha_{k_m} \|d_{1,k_m}\| < L \|d_{2,k_m}\|, \quad \text{for all } k_m. \quad (20)$$

Since sequence f_k is bounded below, it follows from (19) that it follows

$$\|s_{k_m}\| \rightarrow 0 \quad \text{as } k_m \rightarrow \infty. \quad (21)$$

Moreover, since $d_{1k}^T d_{2k} \geq 0$, $\forall k$, from (18) we have

$$\|s_{k_m}\|^2 \geq \alpha_{k_m}^2 (1 - \xi_{k_m})^2 \|d_{1,k_m}\|^2 + \xi_{k_m}^2 \|d_{2,k_m}\|^2. \quad (22)$$

We need to show that

$$d_{1,k_m} = -g_{k_m} = -g(x_{k_m}) \rightarrow 0.$$

Assume the contrary, that is this is not true. For the sake of simplicity, assume that there exists $\tilde{\varepsilon}$ such that

$$\|g_{k_m}\| \geq \tilde{\varepsilon} > 0, \quad \forall k_m. \quad (23)$$

We will show that this leads to a contradiction by considering two possible cases with respect to the sequence $\|d_{2,k_m}\|$. In the first case we assume that this sequence converges to zero, in the second case it does not.

(i) Let

$$\|d_{2,k_m}\| \rightarrow 0 \quad \text{as } k_m \rightarrow \infty.$$

In this case from (20) we have

$$\alpha_{k_m} \|d_{1,k_m}\| = \alpha_{k_m} \|g(x_{k_m})\| \rightarrow 0 \quad \text{as } k_m \rightarrow \infty. \quad (24)$$

Then from uniform continuity of g we obtain that

$$\|g(x_{k_m} - \alpha_{k_m} g(x_{k_m})) - g(x_{k_m})\| \rightarrow 0 \quad \text{as } k_m \rightarrow \infty. \quad (25)$$

From (17) it follows

$$g(x_{k_m} + \alpha_{k_m} d_{1,k_m})^T d_{1,k_m} \geq \sigma g(x_{k_m})^T d_{1,k_m}.$$

Letting $d_{1,k_m} = -g(x_{k_m})$ from the last inequality we obtain

$$[g(x_{k_m} - \alpha_{k_m} g(x_{k_m})) - g(x_{k_m})]^T g(x_{k_m}) \leq (\sigma - 1)g(x_{k_m})^T g(x_{k_m});$$

or

$$\sigma \geq 1 + \frac{[g(x_{k_m} - \alpha_{k_m} g(x_{k_m})) - g(x_{k_m})]^T g(x_{k_m})}{\|g(x_{k_m})\|^2}.$$

Then from (23) and (25) we have $\sigma \geq 1$ that is a contradiction.

(ii) Now we assume that the sequence $\|d_{2,k_m}\|$ does not converge to zero. For the sake of simplicity assume that $\|d_{2,k_m}\| \geq \mu > 0$ for all k_m . Then from (21) and (22) it follows that $\xi_{k_m} \rightarrow 0$ and therefore (24) is satisfied. Then we get a contradiction as in the case of **(i)**.

Therefore (23) leads to a contradiction; that is, $g(x_{k_m}) \rightarrow 0$. □

5. Superlinear Convergence

Theorem 4.1 in the previous section establishes the convergence of gradients $g(x_{k_m})$ that is the stopping criterion for Algorithm 2. In this section, we assume that the sequence of points $\{x_k\}$ generated by the algorithm also converges to some point x^* . In this case we aim to investigate the convergence rate of Algorithm 2.

Denote by $D \subset \mathbb{R}^n$ some convex neighborhood of x^* that contains all elements x_k for sufficiently large k . Since we are interested in the convergence rate of $\{x_k\}$ to x^* , we assume that $x_k \in D$ for all $k = 0, 1, 2, \dots$.

We recall that the function f is assumed to be twice continuously differentiable. In addition we will use the following assumptions.

(AS1) $x^* \in D$ is a strong local minimizer of the function f , (for definition see [18]), with $H(x^*)$ symmetric and positive definite.

(AS2) There is a constant $\gamma \geq 0$ such that

$$\|H(\bar{x}) - H(x)\| \leq \gamma \|\bar{x} - x\|, \forall x, \bar{x} \in D.$$

In the following theorem, we show that under some assumptions Algorithm CGQN, with Step 9.1, is superlinearly convergent; that is,

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0.$$

Theorem 5.1: *Let the function f be twice continuously differentiable and Assumptions AS1-AS2 be satisfied. Consider a sequence $x_k \rightarrow x^*$, $x_k \in D$, $x_{k+1} = x_k + s_k$, that is generated by Algorithm CGQN with a sequence of symmetric bounded and positive definite matrices B_k . Moreover, suppose there is k_0 such that for all $k \geq k_0$: $\cos(\theta_k^0) \geq \delta > 0$, where θ_k^0 is the angle between $d_{1,k}$ and*

$d_{2,k}$; and the iterations s_k in Step 9.1 utilize the formula

$$s_k = \alpha_k(1 - \xi_k)d_{1,k} + \xi_k d_{2,k}.$$

Then $\{x_k\}$ converges superlinearly to x^* if and only if

$$\lim_{k \rightarrow \infty} \frac{\|[B_k - H(x^*)](x_{k+1} - x_k)\|}{\|x_{k+1} - x_k\|} = 0. \quad (26)$$

Proof: The proof of the theorem is based on the following equivalence:

$$\begin{aligned} \lim_{k \rightarrow \infty} \frac{\|[B_k - H(x^*)](x_{k+1} - x_k)\|}{\|x_{k+1} - x_k\|} = 0 &\iff \lim_{k \rightarrow \infty} \frac{\|g_{k+1}\|}{\|x_{k+1} - x_k\|} = 0 \\ &\iff \lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0. \end{aligned} \quad (27)$$

1. First we prove that $\{x_k\}$ converges to x^* superlinearly if the relation (26) holds. If $g(x_k) = 0$ for some k then the theorem is true. Assume that $\|g(x_k)\| > 0$ and $g(x_k) \rightarrow 0$ as $k \rightarrow \infty$. Denote $d(\xi_k) = (1 - \xi_k)d_{1,k} + \xi_k d_{2,k}$, with $d_{1,k} = -g_k$, and let θ_k be the angle between $d(\xi_k)$ and $d_{1,k}$.

Since $\xi_k < 1$ and $d_{1,k}^T d_{2,k} \geq 0$ for all $k \geq k_0$, it is clear that $\theta_k < \theta_k^0$ and therefore by the assumption of the theorem $\cos(\theta_k) > \cos(\theta_k^0) \geq \delta > 0$.

This in particular means that the required inequality in Step 8 is achieved at the first round for η_0 ; that is, ξ_k has the form

$$\xi_k = \frac{1}{1 + \eta_0 |f_k - f_{k-1}|}, \quad \forall k \geq k_0.$$

Since $|f_k - f_{k-1}| \rightarrow 0$ we obtain that

$$\xi_k \rightarrow 1. \quad (28)$$

By assumption, for all $k \geq k_0$, the increments s_k are obtained in Step 9.1 of Algorithm CGQN; that is,

$$x_{k+1} = x_k - \alpha_k(1 - \xi_k)g_k - \xi_k B_k^{-1}g_k. \quad (29)$$

From (29), we have

$$\begin{aligned} [B_k - H(x^*)](x_{k+1} - x_k) &= \\ &= -\alpha_k(1 - \xi_k)B_k g_k - \xi_k g_k - H(x^*)(x_{k+1} - x_k) = \\ &= g_{k+1} - g_k - H(x^*)(x_{k+1} - x_k) - g_{k+1} + (1 - \xi_k)(g_k - \alpha_k B_k g_k). \end{aligned} \quad (30)$$

Taking here the norm and dividing by $\|x_{k+1} - x_k\|$, we obtain

$$\begin{aligned} \frac{\|g_{k+1}\|}{\|x_{k+1} - x_k\|} &\leq \frac{\|[B_k - H(x^*)](x_{k+1} - x_k)\|}{\|x_{k+1} - x_k\|} \\ &+ \frac{\|g_{k+1} - g_k - H(x^*)(x_{k+1} - x_k)\|}{\|x_{k+1} - x_k\|} + \frac{\|(1 - \xi_k)(g_k - \alpha_k B_k g_k)\|}{\|x_{k+1} - x_k\|}. \end{aligned} \quad (31)$$

We note that all the assumptions of Theorems 2.2 and 2.3 are satisfied on the open convex set D . By applying Theorem 2.2 we have

$$\frac{\|g_{k+1} - g_k - H(x^*)(x_{k+1} - x_k)\|}{\|x_{k+1} - x_k\|} \leq \frac{\gamma}{2}(\|x_{k+1} - x^*\| + \|x_k - x^*\|) \rightarrow 0. \quad (32)$$

Now consider the third term on right hand side of (31). Denoting by I the unit matrix, we have

$$\begin{aligned} \frac{\|(1 - \xi_k)(g_k - \alpha_k B_k g_k)\|}{\|x_{k+1} - x_k\|} &= \frac{\|g_k - \alpha_k B_k g_k\|}{\|\alpha_k g_k + \frac{\xi_k}{1 - \xi_k} B_k^{-1} g_k\|} \leq \\ &\frac{\|I - \alpha_k B_k\|}{\|\alpha_k \frac{g_k}{\|g_k\|} + \frac{\xi_k}{1 - \xi_k} B_k^{-1} \frac{g_k}{\|g_k\|}\|}. \end{aligned}$$

Since $\|B_k\|$ is assumed to be bounded for all $k \geq k_0$ the relation $\xi_k \rightarrow 1$ from (28) yields

$$\left\| \frac{\xi_k}{1 - \xi_k} B_k^{-1} \frac{g_k}{\|g_k\|} \right\| \rightarrow \infty \text{ as } k \rightarrow \infty.$$

Indeed, if this is not true, then $\|B_{k_m}^{-1} \frac{g_{k_m}}{\|g_{k_m}\|}\| \rightarrow 0$ for some $k_m \rightarrow \infty$ that contradicts

$$\left\| B_{k_m} \left(B_{k_m}^{-1} \frac{g_{k_m}}{\|g_{k_m}\|} \right) \right\| = 1, \quad \forall k_m.$$

Thus

$$\frac{\|(1 - \xi_k)(g_k - \alpha_k B_k g_k)\|}{\|x_{k+1} - x_k\|} \rightarrow 0 \text{ as } k \rightarrow \infty. \quad (33)$$

Therefore, it follows from (26), (32) and (33) that

$$\lim_{k \rightarrow \infty} \frac{\|g_{k+1}\|}{\|x_{k+1} - x_k\|} = 0. \quad (34)$$

The remaining part of the proof is similar to the proof of Theorem 5.4.3 from [18] which yields $\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0$. This means that the sequence $\{x_k\}$ is convergent to x^* superlinearly.

2. Now suppose that x_k converges to x^* superlinearly. Clearly $g(x^*) = 0$. From the proof of Theorem 5.4.3 [18] it follows that the relation (34) is true. Therefore, from (30) we obtain

$$\begin{aligned} \frac{\| [B_k - H(x^*)](x_{k+1} - x_k) \|}{\|x_{k+1} - x_k\|} &\leq \frac{\|g_{k+1} - g_k - H(x^*)(x_{k+1} - x_k)\|}{\|x_{k+1} - x_k\|} \\ &+ \frac{\|g_{k+1}\|}{\|x_{k+1} - x_k\|} + \frac{\|(1 - \xi_k)(g_k - \alpha_k B_k g_k)\|}{\|x_{k+1} - x_k\|}. \end{aligned} \quad (35)$$

which gives (26) by using (32), (33) and (34). \square

Similar to Theorem 5.1, for Algorithm CGN with Step 9.1 we have the following theorem.

Theorem 5.2: *Let the function f be twice continuously differentiable and Assumptions AS1-AS2 be satisfied. Consider a sequence $x_k \rightarrow x^*$, $x_k \in D$, $x_{k+1} = x_k + s_k$, that is generated by Algorithm CGN. Moreover, suppose there is k_0 such that for all $k \geq k_0$: $\cos(\theta_k^0) \geq \delta > 0$, where θ_k^0 is the angle between $d_{1,k}$ and $d_{2,k}$; and the iterations s_k in Step 9.1 utilize the formula*

$$s_k = \alpha_k(1 - \xi_k)d_{1,k} + \xi_k d_{2,k}.$$

Then $\{x_k\}$ converges to x^* at a superlinear rate.

Proof: From assumption AS2, there is a constant $\gamma \geq 0$ such that

$$\begin{aligned} \frac{\| [H_k - H(x^*)](x_{k+1} - x_k) \|}{\|x_{k+1} - x_k\|} &\leq \frac{\|H_k - H(x^*)\| \|x_{k+1} - x_k\|}{\|x_{k+1} - x_k\|} \\ &= \|H_k - H(x^*)\| \leq \gamma \|x_k - x^*\|. \end{aligned} \quad (36)$$

Since $\{x_k\}$ converges to x^* , we have

$$\lim_{k \rightarrow \infty} \frac{\| [H_k - H(x^*)](x_{k+1} - x_k) \|}{\|x_{k+1} - x_k\|} = 0. \quad (37)$$

The remaining of the proof follows from the proof of Theorem 5.1 with considering H_k instead of B_k . \square

6. Numerical Experiments

In this section, the performance of the proposed algorithms are evaluated by applying them to some unconstrained test problems taken from [15]. Out of 18 unconstrained minimization problems we use 15 problems in the numerical implementations excluding the 3 problem that are global optimization problems. Table 1 gives

a brief description about each test problem, where n is a given integer number by a user. More details can be found in [15].

The group of methods we compare includes our algorithms, algorithms presented by Shi [17], the Newton method (NM), the Quasi-Newton method (QNM) and the gradient method (GM). In the CGN and CGQN algorithms we use two different versions, as described in Steps 9.1 and 9.2. We refer algorithms using Step 9.1 as CGN_1 and $CGQN_1$, and algorithms using Step 9-2 as CGN_2 and $CGQN_2$. From [17], we apply Algorithms 2 and 4, and we refer them as Shi_1 and Shi_2 that are the Newton and Quasi-Newton based methods, respectively.

The termination criteria are the same for all algorithms. Algorithms terminate when either $\|g(x)\| \leq \varepsilon$ or the number of iterations exceeds 500. Parameters in Algorithms CGN and CGQN are chosen as follows: $\varepsilon = 10^{-6}$, $\eta = 10^{-3}$, $\delta = 10^{-3}$, $\vartheta = 1.1$, $\omega = 10^{-10}$, $L = 10^{10}$. Also we select $\rho = 10^{-3}$, $\sigma = 0.9$ for the Wolfe-Powell rule.

The number of iterations (to find the local optimal solutions) used by the algorithms for given initial points are reported in Table 2. In this table, TP stands for test problems, Dim for dimension and IP for initial points. The initial points are taken from [15, 17]. ‘‘AC’’ and ‘‘NC’’ stand for the ‘‘almost convergent’’ and ‘‘not convergent’’, respectively. Convergence means that the method finds the solution x_k where $\|g_k\| < 10^{-6}$, almost convergence means that the method finds a solution x_k where $10^{-6} \leq \|g_k\| \leq 10^{-2}$; otherwise we accept that a method fails to find a solution, that is, it is not convergent.

Based on Table 2, the number of iterations used by Algorithm CGN is, overall, less than those used by other Newton based methods. Especially CGN_1 (using Step 9.1) has the lowest iteration numbers in comparison with other Newton based methods. In the Quasi-Newton based methods, our algorithm ($CGQN$) has found the local optimal solutions using the lowest number of iterations. The gradient method is almost convergent or not convergent in most of the cases.

In order to compare the algorithms with more initial points, we generate 50 random initial points uniformly distributed in $[-10, 10]^n \subset \mathbb{R}^n$ for each test problem 1. Table 3 presents the summary of convergence results in percentage for all test problems. Results from this table demonstrate that Algorithm CGN has the highest convergence rate among Newton based methods. Similarly, Algorithm $CGQN$ has the highest convergence rate among Quasi-Newton based methods.

New algorithms for solving unconstrained optimization problems are presented based on the idea of combining two types of descent directions: the direction of anti-gradient and either the Newton or Quasi-Newton directions. The use of latter directions allows one to improve the convergence rate. Global and superlinear convergence properties of these algorithms are established. Numerical experiments using some unconstrained test problems are reported. Also the proposed algorithms are compared with some existing similar methods using results of experiments. This comparison demonstrates the efficiency of the proposed combined methods.

7. Conclusion

In this paper, we have developed new algorithms which combine the anti-gradient direction with either the Newton direction or the Quasi-Newton direction. We have proved that under some conditions, the first version is both globally and superlinear convergent, while the second version is only globally convergent.

We have carried out a number of experiments using fifteen unconstrained test problems. The numerical results clearly demonstrate the efficiency of proposed algorithms.

References

- [1] J. Biazar, and B. Ghanbari, *A modification on Newtons method for solving systems of nonlinear equations*, World Academy of Science, Engineering and Technology, Vol. 58(4), (2009), pp. 897-901.
- [2] A. G. Buckley, *A combined Conjugate-Gradient Quasi-Newton minimization Algorithm*, Mathematical Programming 15, (1978), pp. 200-210.
- [3] A. G. Buckley, *Extended the relationship between the Conjugate Gradient and BFGS Algorithms*, Mathematical Programming 15, (1978), pp. 343-348.
- [4] A. G. Buckley, *Modified Quasi-Newton methods for solving systems of linear equations*, International Journal of Contemp Mathematics and Science, Vol. 2, no. 15, (2007), pp. 737-744.
- [5] T. De Luca, F. Facchinei, C. Kanzow, *A Semismooth equation approach to the solution of nonlinear complementarity problems*, Mathematical Programming 75, (1996), pp. 407-439.
- [6] R. Fletcher, *Practical methods of optimization*, second edition, Wiley, New York, 1987.
- [7] X. He, and S. Xu, *Process Neural Networks*, Guocheny Shejing Yuan Wangluo, Science Press, 2007
- [8] H.H.H. Homeier, *A modified Newton method with cubic convergence: the multivariate case*, Journal of Computer and Applied Mathematics 169, (2004), pp. 161-169.
- [9] A.E. Kostopoulos, G.S. Androulakis, and T.N. Grapsa, *A new nonmonotone Newton's modification for unconstrained optimization*, Tech. Rep. 09-05, Department of Mathematics, University of Patras, Patras, Greece, 2009.
- [10] J. Kou, *A third-order modification of Newton method for systems of nonlinear equations*, Appl. Math. Comput, Vol. 191, (2007), pp. 117-121.
- [11] D. Luenberger, and Y. Ye, *Linear and Non Linear Programming*, Springer Science+ Business Media, LLC, 2008.
- [12] E.N. Malihoutsaki, I.A. Nikas, and T.N. Grapsa, *Improved Newton's method without direct function evaluations*, J. Comput. Appl.Math. 227 (2009), pp. 206-212.
- [13] A. Malik Hj, M. Mansor, and W. Leong, *A switching criterion in hybrid Quasi-Newton BFGS-steepest descent direction.*, Pertanika J. Sci and Technol 7(2), (1999), pp. 111-123.
- [14] M. Mammadov, and S. Taheri, *A Globally convergent optimization Algorithm for systems of nonlinear equations*, in *In Proceedings of the third global conference on Power Control and Optimization, Gold Coast, Australia*, 2010.
- [15] J.J. More, B.S. Garbow, K.E. Hillstrom, *Testing unconstrained optimization software*, ACM Transactions on Mathematical Software 7(1), (1981), pp. 17-41.
- [16] Y. Shi, *A globalization procedure for solving nonlinear systems of equations*, Numerical Algorithms 12, (1996), pp. 273-286.
- [17] Y. Shi, *Globally convergent Algorithms for unconstrained optimization*, Computational Optimatin and Applications 16, (2000), pp. 295-308.
- [18] W. Sun, Y.X. Yuan, *Optimization theory and methods*, nonlinear programming, Springer, 2006.
- [19] W. Wang, and Y.F. Xu, *A revised Conjugate Gradient projection Algorithm for inequality constrained optimization*, Journal of Computational Mathematics 23, (2005), pp. 217-224.
- [20] J. Yang, *Newton-Conjugate-Gradient methods for solitary wave computations*, Journal of Computational Physics 228, (2009), pp. 7007-7024.
- [21] N. Ye, *The hand book of Data Mining*, Lawrence Erlbaum Associates, Inc, 2003.

Table 1. Some test problems taken from [15], n is a user given integer

| Problem | Function name | Dimension |
|---------|--|-----------|
| P1 | Freudenstein and Roth function | 2 |
| P2 | Box three demential function | 3 |
| P3 | Gaussian function | 3 |
| P4 | Gulf research and development function | 3 |
| P5 | Helical valley function | 3 |
| P6 | Brown and Dennis function | 4 |
| P7 | Wood function | 4 |
| P8 | Biggs EXP6 function | 6 |
| P9 | Watson function | n |
| P10 | Extended Powell singular function | n |
| P11 | Penalty function1 | n |
| P12 | Penalty function2 | n |
| P13 | Trigonometric function | n |
| P14 | Variably dimensioned function | n |
| P15 | Extended Rosenbrock function | n |

Table 2. Number of iterations for 15 test problems obtained by combination of gradient and Newton methods (CGN_1 and CGN_2), combination of gradient and Quasi Newton methods ($CGQN_1$ and $CGQN_2$), Shi's methods [17] (Shi_1, Shi_2), Newton method (NM), Quasi-Newton method (QNM) and gradient method (GM)

| TP | Dim | IP | Newton based methods | | | | Quasi-Newton based methods | | | | GM |
|-----|-----|---------------------------------------|----------------------|---------|---------|----|----------------------------|----------|---------|-----|-----|
| | | | CGN_1 | CGN_2 | Shi_1 | NM | $CGQN_1$ | $CGQN_2$ | Shi_2 | QNM | |
| P1 | 2 | (0.5, -2) | 6 | 8 | 8 | 14 | 9 | 9 | 24 | 17 | AC |
| P1 | 2 | (5, -20) | 7 | 10 | 13 | 24 | 17 | 17 | 26 | 27 | 253 |
| P2 | 3 | (0, 10, 20) | 7 | 7 | 9 | 10 | 23 | 23 | 25 | 41 | AC |
| P2 | 3 | (0, 100, 200) | 12 | 15 | 36 | NC | 14 | 20 | 29 | 31 | 37 |
| P3 | 3 | (0.4, 1, 0) | 2 | 3 | 4 | 5 | 2 | 2 | 6 | 6 | 10 |
| P3 | 3 | (4, 10, 0) | 9 | 15 | 12 | NC | 18 | 20 | 22 | 27 | 180 |
| P4 | 3 | (5, 2.5, 0.15) | 4 | 8 | 11 | 11 | 19 | 34 | 48 | 48 | 66 |
| P4 | 3 | (-5, -2.5, -0.15) | 3 | 5 | 14 | 14 | 13 | 13 | 13 | 13 | 18 |
| P5 | 3 | (-1, 0, 0) | 10 | 10 | 13 | 24 | 17 | 17 | 28 | 22 | AC |
| P5 | 3 | (-10, 0, 0) | 14 | 20 | 19 | NC | 20 | 20 | 37 | 35 | NC |
| P6 | 4 | (25, 5, -5, -1) | 10 | 12 | 13 | 19 | 39 | 44 | 44 | 44 | 320 |
| P6 | 4 | (500, 100, -100, -20) | 16 | 16 | 18 | 18 | 64 | 74 | 67 | 69 | 430 |
| P7 | 4 | (-3, -1, -3, -1) | 21 | AC | AC | NC | 31 | 37 | 81 | 62 | NC |
| P7 | 4 | (3, 1, 3, 1) | 9 | 10 | 11 | 11 | 18 | 27 | 64 | 57 | AC |
| P8 | 6 | (1, 2, 1, 1, 1, 1) | 20 | 23 | 24 | NC | 64 | 64 | 102 | 87 | AC |
| P8 | 6 | (10, 20, 10, 10, 10, 10) | 27 | 30 | 66 | NC | 24 | 26 | 33 | 32 | AC |
| P9 | 6 | (0, 0, 0, 0, 0, 0) | 6 | 11 | 14 | 14 | 28 | 34 | 36 | 38 | AC |
| P9 | 6 | (0.1, 0.1, 0.1, 0.1, 0.1, 0.1) | 8 | 9 | 9 | NC | 29 | 34 | 37 | 35 | AC |
| P9 | 30 | (0, 0, ..., 0) | 7 | 10 | 10 | 14 | 45 | 54 | 59 | 58 | AC |
| P9 | 30 | (0.1, 0.1, ..., 0.1) | 55 | 57 | 66 | 72 | 73 | 94 | 95 | 99 | AC |
| P10 | 4 | (3, -1, 0, 1) | 12 | 12 | 13 | 18 | 20 | 23 | 23 | 25 | AC |
| P10 | 4 | (30, -10, 0, 10) | 17 | 19 | 29 | 18 | 24 | 27 | 36 | 35 | AC |
| P10 | 40 | (3, -1, 0, 1, ..., 3, -1, 0, 1) | 13 | 14 | 20 | 24 | 68 | 73 | 77 | 93 | AC |
| P10 | 40 | (30, -10, 0, 10, ..., 30, -10, 0, 10) | 25 | 28 | 33 | 35 | 53 | 55 | 55 | 59 | AC |
| P11 | 10 | (1, 2, ..., 9, 10) | 5 | 7 | 12 | 10 | 5 | 8 | 14 | 15 | 17 |
| P11 | 10 | (10, 20, ..., 90, 100) | 7 | 9 | 15 | 13 | 13 | 13 | 16 | 18 | 24 |
| P11 | 100 | (1, 2, ..., 90, 100) | 8 | 11 | 26 | NC | 15 | 18 | 19 | 22 | NC |
| P11 | 100 | (10, 20, ..., 900, 1000) | 14 | 14 | 15 | NC | 21 | 23 | 23 | 28 | NC |
| P12 | 10 | (0.5, 0.5, ..., 0.5) | 6 | 6 | 6 | 8 | 10 | 11 | 14 | 14 | 71 |
| P12 | 10 | (5, 5, ..., 5) | 9 | 12 | 17 | 14 | 22 | 22 | 32 | 27 | 48 |
| P12 | 100 | (0.5, 0.5, ..., 0.5) | 22 | 27 | AC | NC | 73 | 89 | 94 | 174 | AC |
| P12 | 100 | (5, 5, ..., 5) | 29 | 30 | AC | NC | 80 | 91 | 102 | 127 | NC |
| P13 | 10 | (0.1, 0.1, ..., 0.1, 0.1) | 10 | 15 | 14 | AC | 18 | 19 | 20 | 21 | 104 |
| P13 | 10 | (1, 1, ..., 1, 1) | 11 | 11 | 14 | 14 | 29 | 39 | 46 | 57 | 123 |
| P13 | 100 | (0.01, 0.01, ..., 0.01, 0.01) | 16 | 17 | 20 | AC | 39 | 56 | 54 | 135 | 137 |
| P13 | 100 | (0.1, 0.1, ..., 0.1, 0.1) | 24 | 23 | 29 | 30 | 440 | 452 | 468 | 480 | 489 |
| P14 | 10 | (0.9, 0.8, ..., 0.1, 0) | 10 | 11 | 14 | 21 | 16 | 20 | 21 | 22 | 27 |
| P14 | 10 | (9, 8, ..., 1, 0) | 9 | 12 | 12 | 26 | 11 | 12 | 22 | 31 | AC |
| P14 | 100 | (0.99, 0.98, ..., 0.01, 0) | 15 | 19 | 24 | 26 | 24 | 24 | 29 | 38 | 57 |
| P14 | 100 | (9.9, 9.8, ..., 0.1, 0) | 29 | 32 | 32 | 36 | 14 | 17 | 20 | 31 | AC |
| P15 | 10 | (-1.2, 1, ..., -1.2, 1) | 10 | 18 | 25 | NC | 81 | 87 | 104 | 95 | AC |
| P15 | 10 | (-12, 10, ..., -12, 10) | 41 | 52 | 71 | NC | 97 | 108 | 115 | 114 | NC |
| P15 | 100 | (-1.2, 1, ..., -1.2, 1) | 16 | 18 | 24 | NC | 327 | 355 | 383 | 411 | AC |
| P15 | 100 | (-12, 10, ..., -12, 10) | 51 | 58 | 62 | NC | 217 | 238 | 241 | 295 | NC |

Table 3. Summary of average convergence results over 15 test problems given in Table 1 with 50 random initial points

| | Algorithm | Convergence | Almost convergence | Non convergence |
|----------------------------|-----------|-------------|--------------------|-----------------|
| Newton based methods | CGN_1 | 98.27 | 1.36 | 0.37 |
| | CGN_2 | 95.72 | 2.82 | 1.46 |
| | Shi_1 | 94.91 | 3.55 | 1.54 |
| | NM | 56.18 | 4.45 | 39.37 |
| Quasi Newton based methods | $CGQN_1$ | 96.72 | 1.91 | 1.37 |
| | $CGQN_2$ | 95.54 | 2.36 | 2.10 |
| | Shi_2 | 94.82 | 2.18 | 3.00 |
| | QNM | 93.12 | 2.09 | 4.79 |
| | GM | 53.45 | 26.27 | 20.28 |