Mishra et al. *Advances in Difference Equations* (2020) 2020:638
https://doi.org/10.1186/s13662-020-03100-2

Advances in Difference Equations
a SpringerOpen Journal

RESEARCH Open Access

On q-BFGS algorithm for unconstrained optimization problems

Shashi Kant Mishra[1], Geetanjali Panda[2], Suvra Kanti Chakraborty[3], Mohammad Esmael Samei[4,6*] and Bhagwat Ram[5]

*Correspondence:
mesamei@gmail.com;
mesamei@basu.ac.ir
[4]Department of Mathematics, Bu-Ali Sina University, 65178 Hamedan, Iran
[6]Department of Medical Research, China Medical University Hospital, China Medical University, Taichung, Taiwan
Full list of author information is available at the end of the article

Abstract

Variants of the Newton method are very popular for solving unconstrained optimization problems. The study on global convergence of the BFGS method has also made good progress. The q-gradient reduces to its classical version when q approaches 1. In this paper, we propose a quantum-Broyden–Fletcher–Goldfarb–Shanno algorithm where the Hessian is constructed using the q-gradient and descent direction is found at each iteration. The algorithm presented in this paper is implemented by applying the independent parameter q in the Armijo–Wolfe conditions to compute the step length which guarantees that the objective function value decreases. The global convergence is established without the convexity assumption on the objective function. Further, the proposed method is verified by the numerical test problems and the results are depicted through the performance profiles.

MSC: Primary 90C30; 65K05; secondary 05A40

Keywords: Unconstrained optimization; BFGS method; q-calculus; Global convergence

1 Introduction

Several numerical methods have been developed extensively for solving unconstrained optimization problems. The gradient descent method is one of the most simplest and commonly used method in the field of the optimization [1]. This method is globally convergent, but suffers from the slow convergence rate as the iterative point approaches the minimizer. In order to improve the convergence rate, optimizers use the Newton method [1]. This method is one of the most popular methods due to its quadratic convergence. A major disadvantage of the Newton method is its slowness or non-convergence for the starting point not being taken close to an optima, and it also requires one to compute the inverse of the Hessian at every iteration, which is rather costly. The components of the Hessian matrix are constructed using the classical derivative, which is positive definite at every iteration. In quasi-Newton methods, instead of computing the actual Hessian, an approximation of the Hessian is considered [1]. These methods use only first derivatives to make an approximation whose computing costs are low.

The Broyden–Fletcher–Goldfarb–Shanno (BFGS) method is the type of quasi-Newton methods for solving unconstrained nonlinear optimization problems which came into existence from the independent work of Broyden [2], Fletcher [3], Goldfarb [4], and Shanno [5]. Since the 1970s the BFGS method became more and more popular and today it is accepted as one of the best quasi-Newton methods. Along these years, many attempts have been made to improve the performance of the quasi-Newton methods [6–18].

The global convergence of the BFGS method have been studied by several authors [5, 12, 19–21] under the convexity assumption on the objective function. An example was given in [22] to fail the standard BFGS method for non-convex functions with inexact line search [12]. A modified BFGS method was developed to converge globally without a convexity assumption on the objective function [23]. In the reference, Li *et al.* concerned with the open problem of whether the BFGS method with inexact line search converges globally when applied to non-convex unconstrained optimization problems [23]. We propose a cautious BFGS update and prove that the method with either a Wolfe-type or an Armijo-type line search converges globally if the function to be minimized has Lipschitz continuous gradients. The q-calculus, particularly known as quantum calculus, has gained a lot of interest in various fields of science, mathematics [24], physics [25], quantum theory [26], statistical mechanics [27] and signal processing [28], etc., where the q-derivative is employed. It is also known as the Jackson derivative, as the concept was first introduced by Jackson [29]; it was further studied in the case of a q-difference equation by Carmichael [30], Mason [31], Adams [32] and Trjitzinsky [33]. The word quantum usually refers to the smallest discrete quantity of some physical property and it comes from the Latin word "quantus", which literally means how many. In mathematics, the quantum calculus is referred as the calculus without limits and it replaces the classical derivative by a difference operator.

A q-version of the steepest descent method was first developed in the field of optimization to solve single objective nonlinear unconstrained problems. The method was able to escape from many local minima and reach the global minimum [34]. The q-LMS (Least Mean Square) algorithm is proposed by employing the q-gradient to compute the secant of the cost function instead of the tangent [28]. The algorithm takes the larger steps towards the optimum solution and achieves a higher convergence rate. An improved version of q-LMS algorithm was developed based on a new class of stochastic q-gradient methods. The proposed approach shows the high convergence rate by utilizing the concept of error correlation energy, and normalization of signal [35]. Global optimization using the q-gradient was further studied in [36], where the parameter q is a dilation that is used to control the degree of localness of the search, solving several multimodal functions. Furthermore, a modified Newton method based on a deterministic scheme using the q-derivative was proposed [37, 38]. Recently, a mathematical package for q-series and partition theory applications has been developed using MATHEMATICA software [39].

A sequence q^k is introduced instead of using a fixed positive number q in the Newton and limited memory BFGS schemes in [37, 40]. After some large value of k, it is obvious that the Hessian becomes almost the same as the exact Hessian of the objective function. The concept of q-gradient, in contrast to the classical gradient in the q-least mean squares algorithm [28], provides extra freedom to control the performance of the algorithm, which we adopt in our proposed method.

In this article, we propose a method using the q-derivative for solving unconstrained optimization problems. This algorithm is different from the classical BFGS algorithm as the search process moves from global in the beginning to local at the end. We utilize an independent parameter $q \in (0,1)$ in Armijo–Wolfe conditions for finding the step length. The proposed algorithm with the Armijo–Wolfe line search is globally convergent for general objective functions. Then we compare the new approach with the existing method.

his paper is organized as follows: In the next section, we give the preliminary idea about the q-calculus. In Sect. 3, we present the q-BFGS (quantum-Broyden–Fletcher–Goldfarb–Shanno) method, using q-calculus. In Sect. 4, the global convergence of proposed algorithm is proved. In Sect. 5, we report some numerical experiments. Finally, we present a conclusion in the last section.

2 Preliminaries

In this section, we present some basic definitions of q-calculus. Given value of $q \neq 1$, we present the q-integer $[n]_q$ [41] by

$$[n]_q = \begin{cases} \frac{1-q^n}{1-q}, & q \neq 1, \\ n, & q = 1, \end{cases}$$

for $n \in \mathbb{N}$. The q-derivative $D_q[f]$ [42] of a function $f : \mathbb{R} \to \mathbb{R}$ is given by

$$D_q[f](x) = \frac{f(x) - f(qx)}{(1-q)x},$$

whenever scalar $q \in (0,1)$, $x \neq 0$ and $D_q[f](0) = f'(0)$ provided $f'(0)$ exists. Note that

$$\lim_{q\to 1} D_q[f](x) = \frac{f(x) - f(qx)}{(1-q)x} = \frac{\mathrm{d}f(x)}{\mathrm{d}x},$$

if f is differentiable. The q-derivative of a function of the form x^n is

$$D_{q,x}\left[x^n\right] = \begin{cases} \frac{1-q^n}{1-q} x^{n-1}, & q \neq 1, \\ n x^{n-1}, & q = 1. \end{cases}$$

Let $f(x)$ be a continuous function on $[a,b]$. Then there exists $\hat{q} \in (0,1)$ and $x \in (a,b)$ [43] such that

$$f(b) - f(a) = \big(D_q[f]\big)(x)(b-a), \tag{1}$$

for $q \in (\hat{q},1) \cup (1,\hat{q}^{-1})$. The q-partial derivative of a function $f : \mathbb{R}^n \to \mathbb{R}$ at $x \in \mathbb{R}^n$ with respect to x_i, where scalar $q \in (0,1)$, is given as [34]

$$D_{q,x_i}[f](x) = \begin{cases} \frac{1}{(1-q)x_i}[f(x_1,x_2,\ldots,x_{i-1},x_i,x_{i+1},\ldots,x_n) \\ \quad - f(x_1,x_2,\ldots,x_{i-1},qx_i,x_{i+1},\ldots,x_n)], & x_i \neq 0, \\ \frac{\partial}{\partial x_i} f(x_1,x_2,\ldots,x_{i-1},0,x_{i+1},\ldots,x_n), & x_i = 0, \\ \frac{\partial}{\partial x_i} f(x_1,x_2,\ldots,x_{i-1},x_i,x_{i+1},\ldots,x_n), & q = 1. \end{cases}$$

We now choose the parameter q as a vector, that is,

$$q = (q_1, \dots, q_i, \dots, q_n)^T \in \mathbb{R}^n.$$

Then the q-gradient vector [34] of f is

$$\nabla_q f(x)^T = \left[D_{q_1,x_1}[f](x) \quad \cdots \quad D_{q_i,x_i}[f](x) \quad \cdots \quad D_{q_n,x_n}[f](x) \right]. \tag{2}$$

Let $\{q_i^k\}$ be a real sequence defined by

$$q_i^{k+1} = 1 - \frac{q_i^k}{(k+1)^2}, \tag{3}$$

for each $i = 1, \dots, n$, where $k \in \{0\} \cup \mathbb{N}$ and $q_i^0 \in (0,1)$ is a fixed starting number, then the sequence $\{q_i^k\}$ converges to $(1, \dots, 1)^T$ as $k \to \infty$ [38]. Thus, the q-gradient reduces to its classical version. For the sake of convenience, we represent the q-gradient vector of f at x^k as

$$g_{q^k}(x^k) = \nabla_{q^k} f(x^k).$$

Example 1 Consider a function $f : \mathbb{R}^2 \to \mathbb{R}$ as $f(x) = x_1^2 x_2 + x_2^2$. Then the q-gradient is given as

$$\nabla_{q^k} f(x)^T = \left[(1 + q_1^k) x_1 x_2 \quad x_1^2 + (1 + q_2^k) x_2 \right].$$

We focus our attention on solving the following unconstrained optimization problems:

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

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously q-derivative. In the next section, we present the q-BFGS algorithm.

3 On q-BFGS algorithm

The BFGS method for solving optimization problems (4) generates a sequence $\{x^k\}$ by the following iterative scheme:

$$x^{k+1} = x^k + \alpha_k d_{q^k}^k, \tag{5}$$

for $k = \{0\} \cup \mathbb{N}$, where α_k is the step length, and $d_{q^k}^k$ is the q-BFGS descent direction obtained by solving the following equation:

$$g_{q^0}^0 = -W^0 d_{q^0}^0,$$

and, for $k \geq 1$, we have

$$g_{q^k}^k = -W^k d_{q^k}^k, \tag{6}$$

where W^k is the q-quasi-Newton update Hessian. The sequence $\{W^k\}$ satisfies the following equation:

$$W^{k+1}s^k = y^k,$$

where $y^k = g_{q^k}^{k+1} - g_{q^k}^k$. We call the famous BFGS (Broyden [2], Fletcher [3], Goldfarb [4], and Shanno [5]) updated formula in the context of q-calculus: q-BFGS. Thus, the Hessian W^k is updated by the q-BFGS formula:

$$W^{k+1} = W^k - \frac{W^k s^k (s^k)^T W^k}{(s^k)^T W^k s^k} + \frac{y^k (y^k)^T}{(y^k)^T s^k}, \tag{7}$$

where $s^k = x^{k+1} - x^k$. A good property of Eq. (7) is that W^{k+1} should inherit the positive definiteness of W^k as long as $(y^k)^T s^k > 0$ and numerically support in the sense of classical BFGS update. The condition $(y^k)^T s^k > 0$ is guaranteed to hold if the step length α_k is determined by the exact or inexact line search methods. For computing the step length, the modified Armijo–Wolfe line search conditions due to the q-gradient are presented as

$$f\left(x^k + \alpha_k d_{q^k}^k\right) \leq f\left(x^k\right) + \sigma_1 \alpha_k \left(d_{q^k}^k\right)^T g_{q^k}^k \tag{8}$$

and

$$\nabla_q f\left(x^k + \alpha_k d_{q^k}^k\right)^T d_{q^k}^k \geq \sigma_2 \left(d_{q^k}^k\right)^T g_{q^k}^k, \tag{9}$$

where $0 < \sigma_1 < \sigma_2 < 1$. The first condition (8) is called the Armijo condition; it ensures a sufficient reduction of the objective function while the second condition (9) is called the curvature condition which ensures the nonacceptance short step length.

The Armijo-type line search does not ensure the condition $(y^k)^T s^k > 0$ and hence W^{k+1} is not positive definite even if W^k is positive definite. In order to ensure positive definiteness of W^{k+1}, the condition $(y^k)^T s^k > 0$ is sometimes used to decide whether or not W^k is updated. More specifically, we present the following update due to [23]:

$$W^{k+1} = \begin{cases} W^k - \frac{W^k s^k (s^k)^T W^k}{(s^k)^T W^k s^k} + \frac{y^k (y^k)^T}{(y^k)^T s^k}, & \frac{(y^k)^T s^k}{\|s^k\|^2} > \epsilon \|g_{q_k}^k\|^\beta, \\ W^k, & \text{otherwise}, \end{cases} \tag{10}$$

where ϵ and β are positive constants.

It is not difficult to see from (10) that the updated matrix W^k is symmetric and positive definite for all k, which is in turn implies that $\{f(x^k)\}$ is a non-increasing sequence when the modified Armijo–Wolfe line search conditions (8) and (9) are used. On the basis of the above theory, we present the following q-BFGS Algorithm 1. In the next section, we shall investigate the global convergence of Algorithm 1.

Algorithm 1 q-BFGS algorithm

Require: Objective function $f : \mathbb{R}^n \to \mathbb{R}$, ϵ is a tolerance for convergence. Choose an initial point $x^0 \in \mathbb{R}^n$, fix $q_i^1 \in (0,1)$, and an initial symmetric positive definite matrix $W^0 \in \mathbb{R}^{n\times n}$.

Ensure: Minimizer x^* encountered with corresponding objective function value $f(x^*)$.

```
Set W^0 = I_n.
for k = 0, 1, 2, ... do
    if ||g^k_{q^k}|| < ε then
        Stop.
    else
        Compute a search direction d^k_{q^k} using (6).
        Find a step length α_k > 0 satisfying (8) and (9).
    end if
    Compute next iterate x^{k+1} = x^k + α_k d^k_{q^k}.
    Update W^{k+1} using (10).
    Set q_i^{k+1} = 1 − q_i^k/(k+1)^2 for all i = 1, ..., n.
end for
```

4 Global convergence

In this section, we present the global convergence of Algorithm 1 under the following two assumptions.

Assumption 1 The objective function $f(x)$ has a lower bound on the level set

$$\Omega = \{x \in \mathbb{R}^n \mid f(x) \leq f(x^0)\},$$

where x^0 is the starting point of Algorithm 1.

Assumption 2 Let function f be a continuously q-derivative on Ω, and there exists a constant $L > 0$ such that $\|g_{q^k}(x) - g_{q^k}(y)\| \leq L(x - y)$, for each $x, y \in \Omega$.

Since $\{f(x^k)\}$ is a non-increasing sequence, it is clear that the sequence $\{x^k\}$ generated by Algorithm 1 is contained in Ω. We present the index set as

$$K := \left\{ j : \frac{(y^j)^T s^j}{\|s^j\|^2} > \epsilon \left\| g^j_{q^j} \right\|^{\beta} \right\}.$$

We can again express (10) as

$$W^{k+1} = \begin{cases} W^k - \frac{W^k s^k (s^k)^T W^k}{(s^k)^T W^k s^k} + \frac{y^k (y^k)^T}{(y^k)^T s^k}, & k \in K, \\ W^k. \end{cases} \tag{11}$$

The following lemma is used to prove the global convergence of Algorithm 1 within the context of q-calculus.

Lemma 3 *Let f be satisfied by Assumption 1 and Assumption 2. Let $\{x^k\}$ be generated by Algorithm 1, with $q_i^k \in (0,1)$, where $i = 1,\dots,n$. If there are positive constants γ_1, and γ_2 such that the inequalities*

1) $\|W^k s^k\| \leq \gamma_1 \|s^k\|$,
2) $(s^k)^T W^k s^k \geq \gamma_2 \|s^k\|^2$,

hold for infinitely many k, then we have

$$\lim_{k\to\infty} \inf \| g_{q^k}(x^k) \| = 0. \tag{12}$$

Proof Since $s^k = \alpha_k d_{q^k}^k$, using Part 1 of this lemma, and (6), we have

$$\| g_{q^k}^k \| \leq \gamma_1 \| d_{q^k}^k \| \tag{13}$$

and

$$\gamma_2 \| d_{q^k}^k \| \leq \| g_{q^k}^k \|. \tag{14}$$

From (13) and (14), we get

$$\gamma_1 \| d_{q^k}^k \| \geq \| g_{q^k}(x^k) \| \geq \gamma_2 \| d_{q^k}^k \|. \tag{15}$$

Substituting $s^k = \alpha_k d_{q^k}^k$ in Part 2, we get

$$\left(d_{q^k}^k\right)^T W^k d_{q^k}^k \geq \gamma_2 \| d_{q^k}^k \|^2. \tag{16}$$

We present the case where the Armijo-type line search (8) is used with backtracking parameter ρ. If $\alpha_k \neq 1$, then we have

$$\sigma_1 \rho^{-1} \alpha_k g_{q^k}(x^k)^T d_{q^k}^k < f\left(x^k + \rho^{-1}\alpha_k d_{q^k}^k\right) - f\left(x^k\right). \tag{17}$$

From the q-mean value theorem, there is a $\theta_k \in (0,1)$ such that

$$f\left(x^k + \rho^{-1}\alpha_k d_{q^k}^k\right) - f\left(x^k\right) = \rho^{-1}\alpha_k g_{q^k}\left(x^k + \theta_k \rho^{-1}\alpha_k d_{q^k}^k\right)^T d_{q^k}^k,$$

that is,

$$\begin{aligned} f\left(x^k + \rho^{-1}\alpha_k d_{q^k}^k\right) - f\left(x^k\right) &= \rho^{-1}\alpha_k g_{q^k}\left(x^k\right)^T d_{q^k}^k \\ &\quad + \rho^{-1}\alpha_k \left(g_{q^k}\left(x^k + \theta_k \rho^{-1}\alpha_k d_{q^k}^k\right) - g_{q^k}\left(x^k\right)\right)^T d_{q^k}^k. \end{aligned}$$

From Assumption 2, we get

$$f\left(x^k + \rho^{-1}\alpha_k d_{q^k}^k\right) - f\left(x^k\right) \leq \rho^{-1}\alpha_k g_{q^k}\left(x^k\right)^T d_{q^k}^k + L\rho^{-2}\alpha_k^2 \| d_{q_k}^k \|^2. \tag{18}$$

From (17) and (18), we get for any $k \in K$

$$\alpha_k \geq \frac{-(1-\sigma_1)\rho g_{q^k}^k (x^k)^T d_{q^k}^k}{L \| d_{q^k}^k \|^2}.$$

Since $-g_{q^k}(x^k) = W^k d_{q^k}^k$,

$$\alpha_k \geq \frac{(1-\sigma_1)\rho(d_{q^k}^k)^T W^k d_{q^k}^k}{L\|d_{q^k}^k\|^2}.$$

Using (16) in the above inequality we get

$$\alpha_k \geq \min\{1, (1-\sigma_1)\gamma_2 L^{-1}\rho\} > 0. \tag{19}$$

We now consider the case where the Wolfe-type line search (9) is used. From (9) and Assumption 2, we get

$$(\sigma_2 - 1)g_{q^k}(x^k)^T d_{q^k}^k \leq \left(g_{q^k}(x^k + \alpha_k d_{q^k}^k) - g_{q^k}(x^k)\right)^T d_{q^k}^k \leq L\alpha_k \left\|d_{q^k}^k\right\|^2.$$

This implies that

$$\alpha_k \geq \frac{(\sigma_2 - 1)g_{q^k}(x^k)^T d_{q^k}^k}{L\|d_{q^k}^k\|^2}.$$

Since $-g_{q^k}^k = W^k d_{q^k}^k$,

$$\alpha_k \geq \frac{(1-\sigma_2)(d_{q^k}^k)^T W^k d_{q^k}^k}{L\|d_{q^k}^k\|^2}.$$

Since $W^k d_{q^k}^k \geq \gamma_2 \|d_{q^k}^k\|^2$,

$$\alpha_k \geq \min\{1, (1-\sigma_2)\gamma_2 L^{-1}\rho\} > 0. \tag{20}$$

The inequalities (19) together with (20) show that $\{\alpha_k\}_{k\in K}$ is bounded below away from zero when we use the Armijo–Wolfe line search conditions. Moreover,

$$\begin{aligned}\sum_{k=0}^{\infty}\left[f(x^k) - f(x^{k+1})\right] &= \lim_{j\to\infty}\sum_{k=1}^{j}\left[f(x^k) - f(x^{k+1})\right] \\ &= f(x^1) - \lim_{j\to\infty} f(x^j),\end{aligned}$$

that is,

$$\sum_{k=0}^{\infty}\left[f(x^k) - f(x^{k+1})\right] = f(x^1) - f(x^*),$$

this gives

$$\sum_{k=1}^{\infty}\left[f(x^k) - f(x^{k+1})\right] < \infty.$$

This together with (9) gives

$$-\sum_{k=1}^{\infty} \alpha_k g_{q^k}^T d_{q^k}^k < \infty.$$

Since $g_{q^k}^k = -W^k d_{q^k}^k$,

$$\lim_{k\to\infty} \left(d_{q^k}^k\right)^T W^k d_{q^k}^k = -\lim_{k\to\infty} g_{q^k}\left(x^k\right)^T d_{q^k}^k \to 0.$$

This together with (15) and (16) implies (12). □

Lemma 3 indicates that to prove the global convergence of Algorithm 1, it suffices to show that there are positive constants γ_1 and γ_2 such that Part 1 and Part 2 hold for infinitely many k. For this purpose, we require the following lemma which may be proved in the light of [20, Theorem 2.1].

Lemma 4 *If there are positive constants γ_1 and γ_2 such that, for each $k \geq 0$,*

$$\frac{\|y^k\|^2}{(s^k)^T y^k} \leq M, \qquad \frac{(s^k)^T y^k}{\|s^k\|^2} \geq m, \tag{21}$$

then there exist constants γ_1 and γ_2 such that, for any positive integer t, Part 1 and Part 2 of Lemma 3 hold for at least $\lceil \frac{t}{2} \rceil$ values of $k \in \{1, \dots, t\}$.

From Lemma 3 and Lemma 4, we now prove the global convergence for Algorithm 1.

Theorem 5 *Let f satisfy Assumption 1 and Assumption 2, and $\{x^k\}$ be generated by Algorithm 1. Then Eq. (12) holds.*

Proof If K is finite then W^k remains constant after a finite number of iterations. Since W^k is symmetric and positive definite for each k, it is obvious that there are constants γ_1, and γ_2 such that Part 1 and Part 2 of Lemma 3 hold for all k sufficiently large. If K is infinite, then for the sake of contradiction (12) is not true. Then there exists a positive constant δ such that for all k

$$\left\|g_{q^k}^k\right\| > \delta. \tag{22}$$

Since $(y^k)^T s^k \geq \epsilon \delta^\alpha \|s^k\|^2$,

$$\frac{1}{(y^k)^T s^k} \leq \frac{1}{\epsilon \delta^\alpha \|s^k\|^2}.$$

We know that $\|y^k\|^2 \leq L^2 \|s^k\|^2$. Thus, we get

$$\frac{\|y^k\|^2}{(y^k)^T s^k} \leq \frac{\|y^k\|^2}{\epsilon \delta^\alpha \|s^k\|^2}.$$

From Assumption 2, we get

$$\frac{\|y^k\|^2}{(y^k)^T s^k} \leq M,$$

for each $k \geq 0$, where $M = \frac{L^2}{\epsilon\delta^\alpha}$. Applying Lemma 4 to the matrix subsequence $\{W^k\}_{k\in K}$, we conclude that Part 1 and Part 2 of Lemma 3 hold for infinitely many k. There exists a subsequence of $\{x^k\}$ converging to the q-critical point of (4). As $k \to \infty$, since q^k approaches $(1, 1, \dots, 1)^T$, a q-critical point eventually approximates a critical point. If the objective function f is convex then every local minimum point is a global minimum point. Since the sequence $\{f(x^k)\}$ converges, every accumulation point of $\{x^k\}$ is a global optimal solution of (4). Then Lemma 3 completes the proof. □

The above theorem proved the global convergence of q-BFGS algorithm without the convexity assumption on the objective function.

5 Numerical experiments

This section reports some numerical experiments with Algorithm 1. We teated on some test problems taken from [44]. Our numerical experiments are performed on a Laptop with Intel(R) Core(TM) CPU (i3-4005U@1.70 GHz) and 4 GB RAM, with MATLAB (2017a).

Table 1 Computational details of Example 2 for sequence $\{q^k\}$

k	q^k	$x = (2,3)^T$			
		$f(x)$	$f(q^k x)$	$g_{q^k}(1)$	$g_{q^k}(2)$
1	$(0.3200, 0.3200)^T$	8.4877	7.3482	4.0387	0.5585
2	$(0.9200, 0.9200)^T$	8.4877	8.4043	6.8282	0.3474
3	$(0.9060, 0.9060)^T$	8.4877	8.3889	6.7358	0.3501
4	$(0.9535, 0.9535)^T$	8.4877	8.4401	7.0561	0.3413
5	$(0.9669, 0.9669)^T$	8.4877	8.4540	7.1500	0.3390
6	$(0.9765, 0.9765)^T$	8.4877	8.4639	7.2183	0.3373
7	$(0.9823, 0.9823)^T$	8.4877	8.4698	7.2598	0.3363
8	$(0.9862, 0.9862)^T$	8.4877	8.4738	7.2881	0.3357
9	$(0.9890, 0.9890)^T$	8.4877	8.4766	7.3080	0.3352
10	$(0.9910, 0.9910)^T$	8.4877	8.4786	7.3226	0.3349
11	$(0.9925, 0.9925)^T$	8.4877	8.4801	7.3336	0.3346
12	$(0.9936, 0.9936)^T$	8.4877	8.4813	7.3420	0.3344
13	$(0.9945, 0.9945)^T$	8.4877	8.4822	7.3487	0.3343
14	$(0.9952, 0.9952)^T$	8.4877	8.4829	7.3541	0.3341
15	$(0.9958, 0.9958)^T$	8.4877	8.4835	7.3584	0.3340
16	$(0.9963, 0.9963)^T$	8.4877	8.4840	7.3620	0.3339
17	$(0.9967, 0.9967)^T$	8.4877	8.4844	7.3650	0.3339
18	$(0.9971, 0.9971)^T$	8.4877	8.4847	7.3675	0.3338
19	$(0.9974, 0.9974)^T$	8.4877	8.4850	7.3697	0.3338
20	$(0.9976, 0.9976)^T$	8.4877	8.4853	7.3715	0.3337
21	$(0.9978, 0.9978)^T$	8.4877	8.4855	7.3731	0.3337
22	$(0.9980, 0.9980)^T$	8.4877	8.4857	7.3745	0.3337
23	$(0.9982, 0.9982)^T$	8.4877	8.4859	7.3757	0.3336
24	$(0.9983, 0.9983)^T$	8.4877	8.4860	7.3768	0.3336
25	$(0.9985, 0.9985)^T$	8.4877	8.4861	7.3777	0.3336
26	$(0.9986, 0.9986)^T$	8.4877	8.4862	7.3785	0.3336
27	$(0.9987, 0.9987)^T$	8.4877	8.4863	7.3793	0.3336
28	$(0.9988, 0.9988)^T$	8.4877	8.4864	7.3800	0.3335
29	$(0.9989, 0.9989)^T$	8.4877	8.4865	7.3806	0.3335
30	$(0.9989, 0.9989)^T$	8.4877	8.4866	7.3811	0.3335

Table 2 Computational details of Example 2 for sequence $\{q^k\}$

k	q^k	$x = (-4, 5)^T$			
		$f(x)$	$f(q^k x)$	$g_{q^k}(1)$	$g_{q^k}(2)$
1	$(0.3200, 0.3200)^T$	1.6278	0.4883	0.095486	0.335128
2	$(0.9200, 0.9200)^T$	1.6278	1.5444	0.021585	0.208454
3	$(0.9060, 0.9060)^T$	1.6278	1.5290	0.022236	0.210039
4	$(0.9535, 0.9535)^T$	1.6278	1.5802	0.020129	0.204796
5	$(0.9669, 0.9669)^T$	1.6278	1.5941	0.019582	0.203381
6	$(0.9765, 0.9765)^T$	1.6278	1.6040	0.019203	0.202385
7	$(0.9823, 0.9823)^T$	1.6278	1.6099	0.018979	0.201791
8	$(0.9862, 0.9862)^T$	1.6278	1.6139	0.01883	0.201392
9	$(0.9890, 0.9890)^T$	1.6278	1.6166	0.018726	0.201113
10	$(0.9910, 0.9910)^T$	1.6278	1.6187	0.018651	0.200910
11	$(0.9925, 0.9925)^T$	1.6278	1.6202	0.018595	0.200758
12	$(0.9936, 0.9936)^T$	1.6278	1.6213	0.018552	0.200642
13	$(0.9945, 0.9945)^T$	1.6278	1.6223	0.018518	0.200550
14	$(0.9952, 0.9952)^T$	1.6278	1.6230	0.018491	0.200477
15	$(0.9958, 0.9958)^T$	1.6278	1.6236	0.018469	0.200417
16	$(0.9963, 0.9963)^T$	1.6278	1.6241	0.018451	0.200368
17	$(0.9967, 0.9967)^T$	1.6278	1.6245	0.018436	0.200327
18	$(0.9971, 0.9971)^T$	1.6278	1.6248	0.018423	0.200293
19	$(0.9974, 0.9974)^T$	1.6278	1.6251	0.018412	0.200263
20	$(0.9976, 0.9976)^T$	1.6278	1.6254	0.018403	0.200238
21	$(0.9978, 0.9978)^T$	1.6278	1.6256	0.018395	0.200217
22	$(0.9980, 0.9980)^T$	1.6278	1.6258	0.018388	0.200198
23	$(0.9982, 0.9982)^T$	1.6278	1.6259	0.018382	0.200181
24	$(0.9983, 0.9983)^T$	1.6278	1.6261	0.018377	0.200167
25	$(0.9985, 0.9985)^T$	1.6278	1.6262	0.018372	0.200154
26	$(0.9986, 0.9986)^T$	1.6278	1.6263	0.018368	0.200142
27	$(0.9987, 0.9987)^T$	1.6278	1.6264	0.018364	0.200132
28	$(0.9988, 0.9988)^T$	1.6278	1.6265	0.018361	0.200123
29	$(0.9989, 0.9989)^T$	1.6278	1.6266	0.018358	0.200115
30	$(0.9989, 0.9989)^T$	1.6278	1.6267	0.018355	0.200108

We used the condition

$$\|g_q(x^k)\| \leq 10^{-6},$$

as the stopping criterion. The program stops if the total iteration number is larger than 400. For each problem we choose the initial matrix $W^0 = I_n$, where I_n is an identity matrix. First, we find the q-gradient of the following problem when the parameter q is not fixed.

Example 2 Consider a function $f : \mathbb{R}^2 \to \mathbb{R}$ such that

$$f(x) = e^{x_1} + \log x_2.$$

We need to find the q-gradient vector at $x = (2, 3)^T$ and $x = (-4, 5)^T$. This function uses the sequence $\{q^k\}$ with an initial vector of

$$q^0 = (0.32, 0.32)^T.$$

In fact, Tables 1 and 2 show the computational values of $f(x)$, $f(q^k x)$ and the q-gradient, where $g_{q^k}(1)$ and $g_{q^k}(2)$ are the first and second component of q-gradient. We obtain the

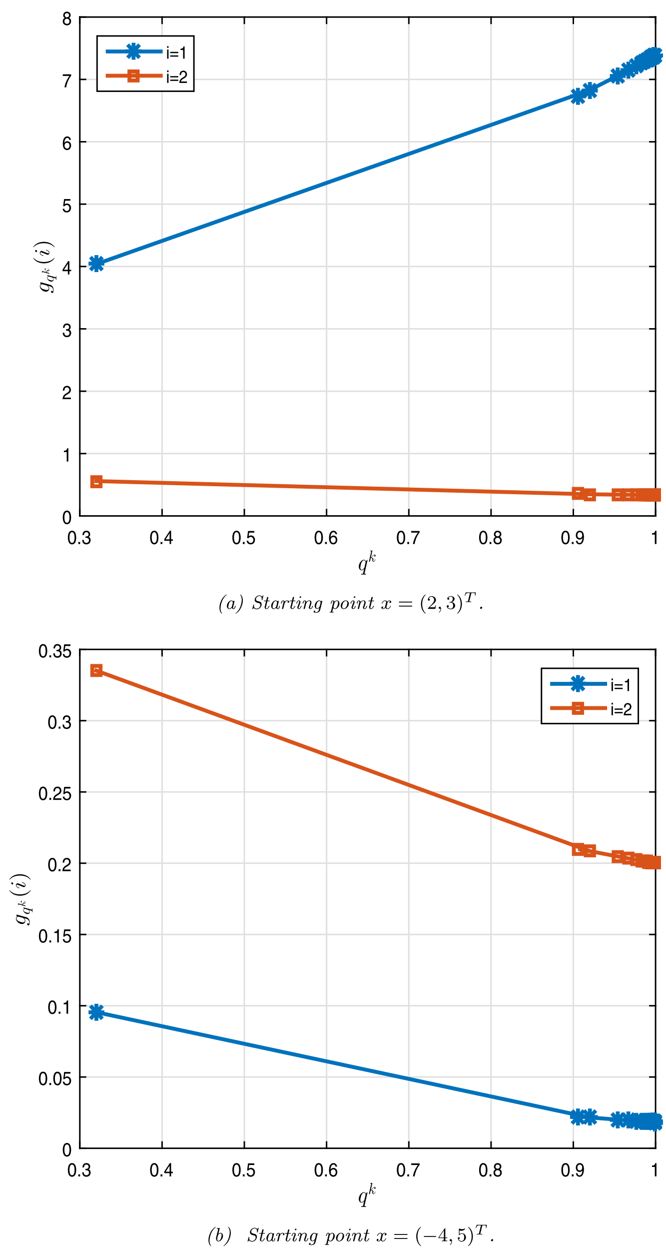


Figure 1 Graphical representation of q-gradient of Tables 1 and 2

q-gradient as

$$g_{q^{30}}(x) = \begin{bmatrix} 7.3811 \\ 0.3335 \end{bmatrix}, \qquad g_{q^{30}}(x) = \begin{bmatrix} 0.018355 \\ 0.200108 \end{bmatrix},$$

for $x = (2,3)^T$ and $x = (-4,5)^T$, respectively, where

$$q^{30} = (0.9989, 0.9989),$$

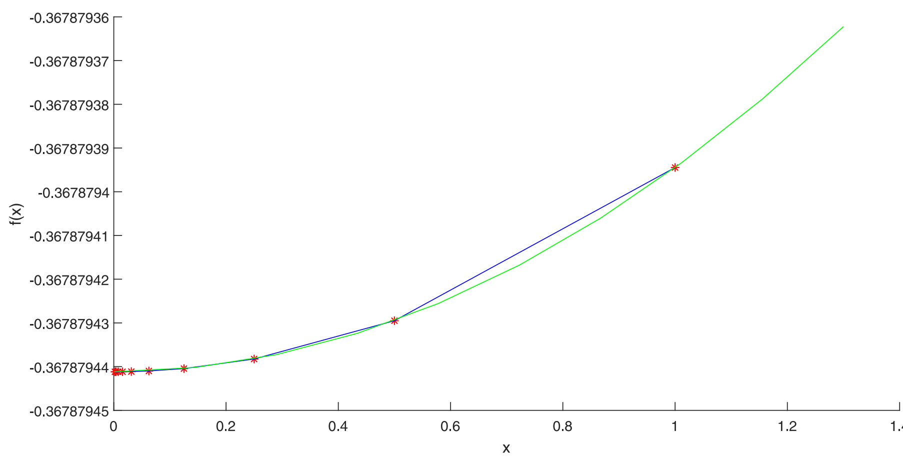


Figure 2 Iteration points of Example 3

at $k = 30$. From Fig. 1, one can observe that q_1^k and $q_2^k \in (0,1)$ for $k = 1,\dots,30$. As $q^k \to (1,1)$, the q-gradient reduces to the classical gradient. For this case, we have

$$g(x) = \begin{bmatrix} 7.3891 \\ 0.3333 \end{bmatrix}, \qquad g(x) = \begin{bmatrix} 0.018315 \\ 0.200000 \end{bmatrix},$$

for $x = (2,3)^T$ and $x = (-4,5)^T$, respectively.

Example 3 Consider an unconstrained objective function [45] $f : \mathbb{R} \to \mathbb{R}$ such that

$$f(x) = -xe^{-x}.$$

This function has the unique minimizer $x^* = 1$. We run Algorithm 1 with a starting point $x^0 = 9.0$ and get the minimum function value

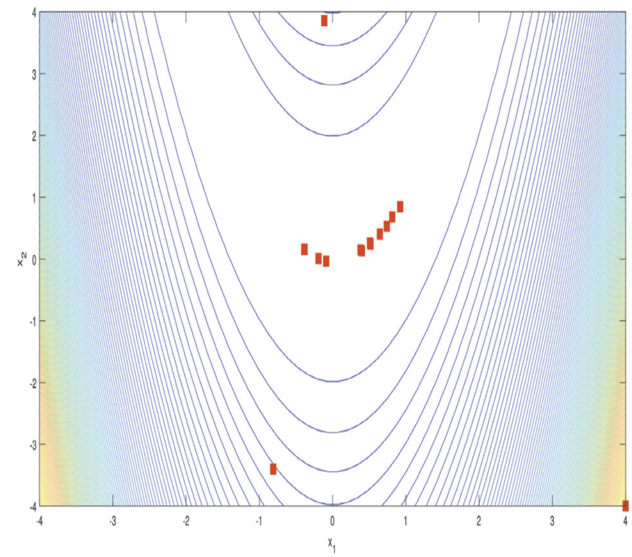
$$f(x^*) = -3.67875,$$

at minimizer $x^* = 1.0$ in 7 iterations, which can be seen in Fig. 2. With different starting points 15, 17 and 19, the algorithm converges to the solution point 1.00, 0.9999 and 0.9998 in iterations 7, 4 and 5, respectively.

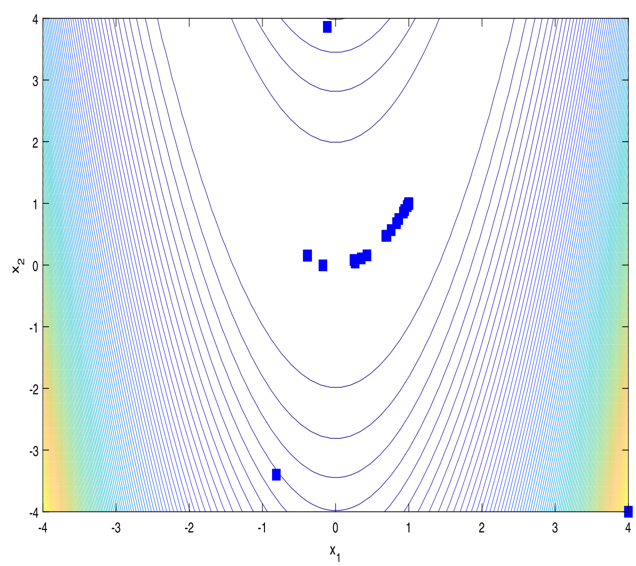
Example 4 Consider an unconstrained optimization function $f : \mathbb{R}^2 \to \mathbb{R}$ such that

$$f(x_1,x_2) = 100\left(x_2 - x_1^2\right)^2 + (1 - x_1)^2.$$

The Rosenbrock function is a non-convex function, introduced by Rosenbrock in 1960. We consider this function to measure the performance of Algorithm 1. In this case, a



(a) Sequence of iterative points generated due to q-gradient



(b) Sequence of iterative points generated due to classical gradient

Figure 3 Comparison of Algorithm 1 with [23] on Example 4

starting point

$$x^0 = (4, -4)^T,$$

is taken. The function converges in 20 iterations to get the minimum function value

$$f(x^*) = 0.0039936,$$

at minimizer

$$x^* = (0.9822, 0.9587)^T,$$

while we run the Algorithm used in the methodology of [23], the function converges in 29 iterations to get the minimum function value

$$f(x^*) = 1.1145 \times 10^{-15},$$

at minimizer

$$x^* = (0.9998, 0.9996)^T,$$

in 29 iterations that are shown in Fig. 3. Of course, Fig. 3(a) shows that our proposed method takes larger steps to converge due to the q-gradient.

With different starting points, we compare our algorithm with [23] on the Rosenbrock function. The numerical results is shown in Tables 3 and 4 where the columns '*it*', '*fe*', and '*ge*' indicate the total numbers of iterations, the total number of function evaluations, and the total number of gradient evaluations, respectively. Note the total number of q-gradient evaluations for q-BFGS and total number of gradient evaluations for BFGS use the same notation.

Table 3 Numerical results of Example 4

Starting point	q-BFGS algorithm				
	it	*fe*	*ge*	x^*	$f(x^*)$
$(4,3)^T$	31	199	33	$(1.047, 1.088)^T$	0.0112
$(-3,1)^T$	13	98	16	$(0.914, 0.852)^T$	0.013401
$(-1,3)^T$	28	179	32	$(1.016, 1.018)^T$	0.009
$(-1.5,3.7)^T$	24	158	29	$(0.879, 0.753)^T$	0.007
$(-1,4)^T$	19	123	19	$(0.967, 0.939)^T$	2.39e−03
$(1,-1)^T$	16	122	18	$(1.037, 1.066)^T$	0.010
$(-4,2)^T$	23	151	27	$(0.928, 0.869)^T$	0.000
$(-1,-4)^T$	17	116	18	$(0.830, 0.691)^T$	0.015
$(-2,2)^T$	16	113	19	$(0.721, 0.496)^T$	0.001
$(-5,6)^T$	27	178	31	$(0.908, 0.823)^T$	0.006
$(-3,6)^T$	32	199	36	$(0.975, 0.948)^T$	0.002
$(4,-5)^T$	18	125	21	$(0.859, 0.721)^T$	0.003
$(4,-7)^T$	20	142	22	$(1.046, 1.084)^T$	0.011
$(-5,-3)^T$	14	111	16	$(0.949, 0.897)^T$	0.005
$(4,-5.6)^T$	14	109	16	$(0.976, 0.959)^T$	0.004
$(-8,2)^T$	3	41	4	$(1.000, 2.563)^T$	0.008
$(-5,7)^T$	26	165	27	$(0.991, 0.964)^T$	0.006
$(-2,6)^T$	33	199	37	$(0.975, 0.941)^T$	0.008
$(1,-5)^T$	19	152	20	$(1.057, 1.113)^T$	0.014
$(-3,-4)^T$	16	115	17	$(0.933, 0.873)^T$	0.000
$(8,1)^T$	21	149	27	$(0.723, 0.499)^T$	0.000
$(3,-7)^T$	12	90	13	$(0.885, 0.789)^T$	0.006
$(4,-5)^T$	18	125	21	$(0.859, 0.721)^T$	0.003
$(-5,-2)^T$	17	117	18	$(0.940, 0.886)^T$	0.002
$(4,-6)^T$	16	119	19	$(1.119, 1.238)^T$	0.012
$(3,-4)^T$	12	96	13	$(0.993, 0.978)^T$	0.003
$(4,-4)^T$	15	108	16	$(0.815, 0.6813)^T$	0.011

Table 4 Numerical results of Example 4

Starting point	BFGS algorithm [23]				
	it	*fe*	*ge*	x^*	$f(x^*)$
$(4,3)^T$	31	173	33	$(0.999, 0.999)^T$	7.75e–16
$(-3,1)^T$	19	115	22	$(0.999, 0.999)^T$	1.01e–16
$(-1,3)^T$	30	172	35	$(1.000, 1.000)^T$	6.33e–17
$(-1.5,3.7)^T$	30	171	35	$(0.999, 0.999)^T$	4.14e–16
$(-1,4)^T$	25	142	29	$(0.999, 0.999)^T$	1.62e–16
$(1,-1)^T$	15	91	17	$(1.000, 1.000)^T$	8.33e–17
$(-4,2)^T$	27	158	33	$(0.999, 0.999)^T$	1.87e–15
$(-1,-4)^T$	22	132	26	$(1.000, 1.000)^T$	3.02e–16
$(-2,2)^T$	27	149	31	$(0.999, 0.998)^T$	8.51e–15
$(-5,6)^T$	30	171	34	$(0.999, 0.999)^T$	4.02e–16
$(-3,6)^T$	25	149	29	$(0.999, 0.999)^T$	7.89e–16
$(4,-5)^T$	22	126	24	$(0.999, 0.999)^T$	3.90e–16
$(4,-7)^T$	21	128	24	$(1.000, 1.000)^T$	1.22e–15
$(-5,-3)^T$	22	125	23	$(0.999, 0.999)^T$	1.08e–15
$(4,-5.6)^T$	24	141	28	$(0.999, 0.999)^T$	3.96e–16
$(-8,2)^T$	11	74	13	$(1.000, 1.000)^T$	4.03e–16
$(-5,7)^T$	30	175	37	$(1.000, 1.000)^T$	2.70e–16
$(-2,6)^T$	36	192	41	$(0.999, 0.999)^T$	7.46e–17
$(1,-5)^T$	16	97	20	$(1.000, 1.000)^T$	2.94e–16
$(-3,-4)^T$	26	151	30	$(0.999, 0.999)^T$	1.76e–16
$(8,1)^T$	31	178	34	$(0.999, 0.999)^T$	7.60e–16
$(3,-7)^T$	12	77	15	$(0.999, 0.999)^T$	1.30e–16
$(4,-5)^T$	22	126	24	$(0.999, 0.999)^T$	3.90e–16
$(-5,-2)^T$	21	125	24	$(1.000, 1.000)^T$	3.63e–16
$(4,-6)^T$	22	134	26	$(0.999, 0.999)^T$	7.36e–16
$(3,-4)^T$	18	110	23	$(0.999, 0.999)^T$	3.93e–16
$(4,-4)^T$	24	140	25	$(0.999, 0.999)^T$	3.97e–16

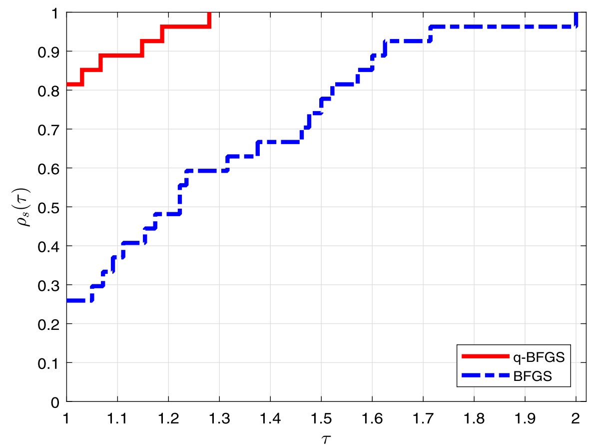
Dolan and Moré [46] presented an appropriate technique to demonstrate the performance profiles, which is a statistical process. The performance ratio is presented as

$$\rho_{p,s} = \frac{r_{(p,s)}}{\min\{r_{(p,s)} : 1 \leq r \leq n_s\}}, \tag{23}$$

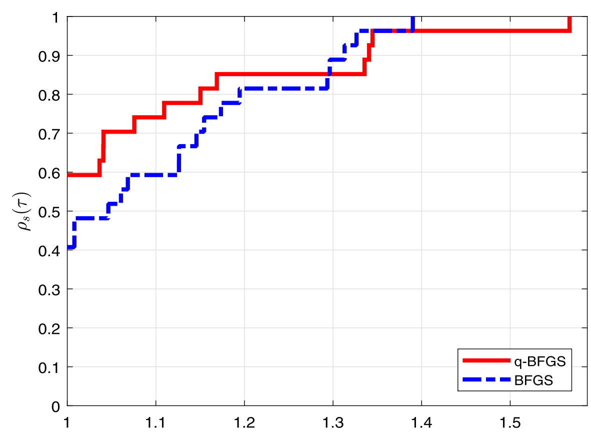
where $r_{(p,s)}$ refers to the iteration, function evaluations and q-gradient evaluations, respectively for solver s spent on problem p and n_s refers to the number of problems in the model test. The cumulative distribution function is expressed as

$$P_s(\tau) = \frac{1}{n_p} \operatorname{size}\{p \in \rho_{(p,s)} \leq \tau\}, \tag{24}$$

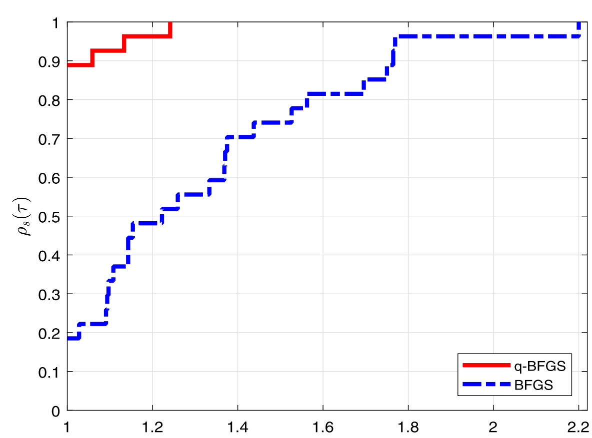
where $P_s(\tau)$ is the probability that a performance ratio $\rho_{(p,s)}$ is within a factor of τ of the best possible ratio. That is, for a subset of the methods being analyzed, we plot the fraction $\rho_s(\tau)$ of problems for which any given method is within a factor τ of the best. We use this tool to show the performance of Algorithm 1. Here, Fig. 4(a), Fig. 4(b) and Fig. 4(c) show that q-BFGS method solves about 82%, 59% and 89% of Rosenbrock test problem with the least number of iterations, function evaluations and gradient evaluations, respectively.



(a) Performance Profile based on number of iterations.



(b) Performance Profile based on number of function evaluations.



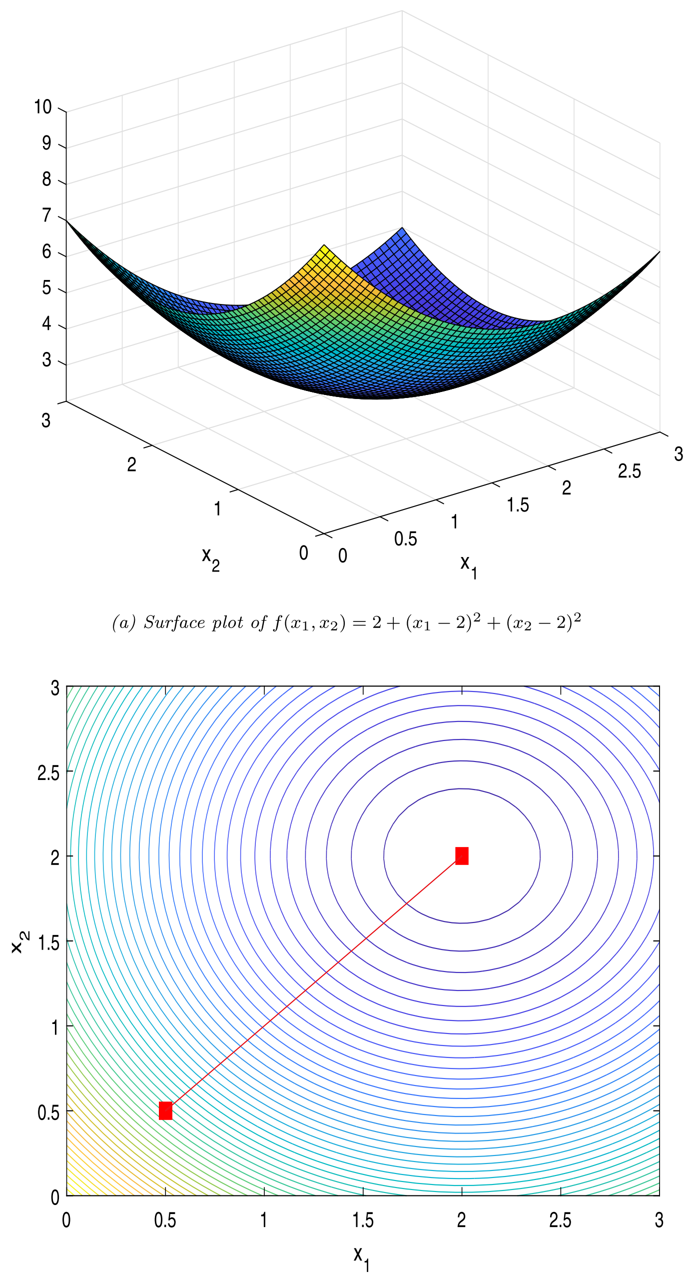
(c) Performance Profile based on number of gradient evaluations.

Figure 4 Performance profiles based on the number of iterations, the number of function evaluations and the number of gradient evaluations given in Tables 3 and 4

Example 5 Consider an unconstrained optimization problem $f : \mathbb{R}^2 \to \mathbb{R}$ such that

$$f(x_1, x_2) = 2 + (x_1 - 2)^2 + (x_2 - 2)^2.$$

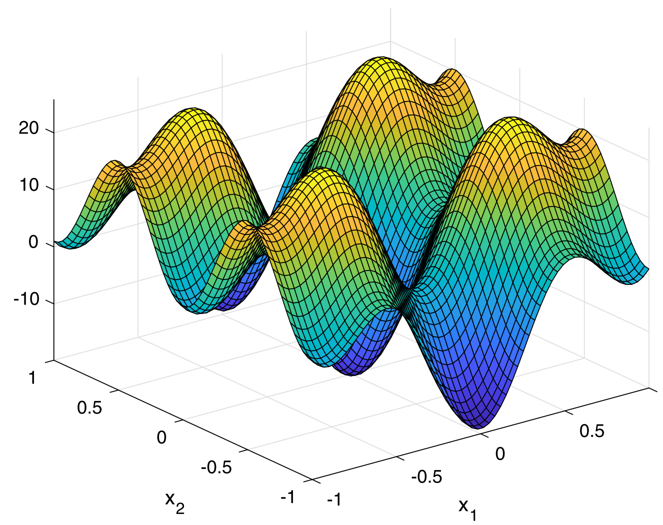
With a starting point $x^0 = (0.5, 0.5)^T$, the function converges to $x^* = (2, 2)^T$ that is shown in Fig. 5(a). The global minima of this function can also be observed in Fig. 5(b).



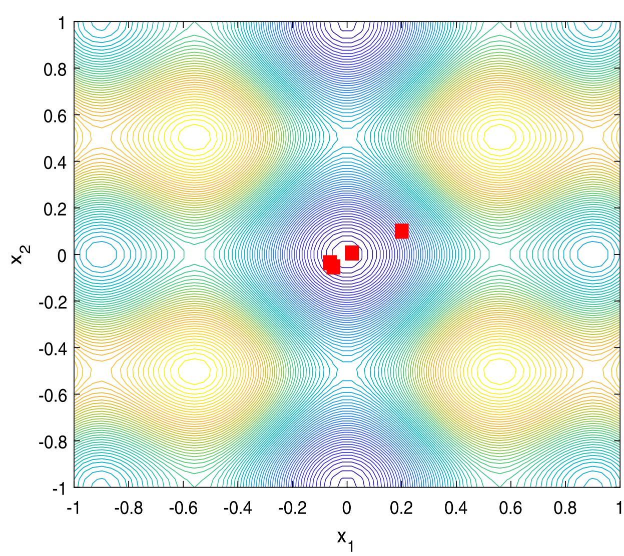
(a) Surface plot of $f(x_1, x_2) = 2 + (x_1 - 2)^2 + (x_2 - 2)^2$

(b) Global minima of $f(x_1, x_2) = 2 + (x_1 - 2)^2 + (x_2 - 2)^2$

Figure 5 Visualization of Example 5



(a) Several local minima of $f(x_1, x_2) = 20 + x_1^2 + x_2^2 - 10(\cos 2\pi x_1) + \cos 2\pi x_2$.



(b) Global minima of $f(x_1, x_2) = 20 + x_1^2 + x_2^2 - 10(\cos 2\pi x_1) + \cos 2\pi x_2$.

Figure 6 Visualization of Example 6

Table 5 Numerical results of Example 6

k	*fe*	*qe*	$f(x^*)$
0	1	1	8.86966
1	7	2	1.05224
2	19	3	1.01602
3	24	4	0.0702737
4	29	5	0.0055612

Table 6 Set of Test Problems

Problem number	Problem's name	Starting Point
1	Rastrigin	$(0.2, 0.1)^T$
2	mvfChichinadze	$(4.5, 0.52)^T$
3	Price Function	$(1, -2)^T$
4	Beale	$(3, 1.5)^T$
5	mvfBohachevsky1	$(0.1, 0.2)^T$
6	mvfBohachevsky2	$(1, -5.8)^T$
7	Box-Betts exponential	$(-8, -33, -1)^T$
8	Branin	$(9.3, 3)^T$
9	Three-Hump Camel	$(-3.4, 0)^T$
10	Six Hump Camel	$(-0.6, 2)^T$
11	Chichinadze	$(5, 7, 0.4)^T$
12	Colville	$(0.7, 0.413)^T$
13	Easom	$(3, 3.8)^T$
14	Exp2	$(1, 10)^T$
15	Goldstein–Price	$(2, -1.2)^T$
16	Griewank10	$(2, -1.2)^T$
17	Hansen	$(4.9, 5.2)^T$
18	Hartman 3D	$(-1, 2, 1)^T$
19	SHEKEL	$(3, 4, 4, 4)^T$

Example 6 Consider the following Rastrigin function:

$$f(x) = 20 + x_1^2 + x_2^2 - 10(\cos 2\pi x_1) + \cos 2\pi x_2.$$

The Rastrigin function is a non-convex. The visualization of in an area from −1 to 1 is shown in Fig. 6(a) with many local minima, and the global optimum at $(0, 0)^T$ in an area from −1 to 1 is shown in Fig. 6(b) with successive iterative points. The numerical results of this function is shown in Table 5.

We have used the 19 test problems as shown in Table 6 with attributes problem number, problem's name and starting point, respectively. Table 7 shows the computational results for q-BFGS and BFGS method [23] on small scale test problems. Of course, Fig. 7(a), Fig. 7(b) and Fig. 7(c) show that the q-BFGS method solves about 95%, 79% and 90% of the test problems with the least number of iterations, function evaluations and gradient evaluations, respectively. Therefore, this is concluded that the q-BFGS performs better than the BFGS of [23] and improves the performance in fewer iterations, function evaluations, and gradient evaluations, respectively.

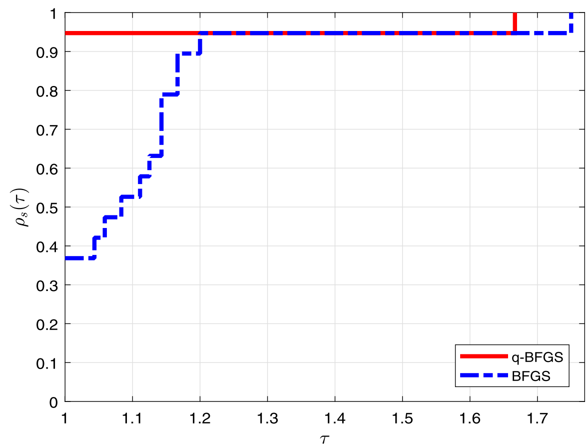
6 Conclusion

We have proposed a q-BFGS update and shown that the method converges globally with the Armijo–Wolfe line search conditions. The variant of the proposed method behaves like the classical BFGS method in limiting case where the existence of second-order partial derivatives at every point is not required. First-order q-differentiability of the function is sufficient to prove the global convergence of the proposed method. The q-gradient enables the q-BFGS quasi-Newton search process to be carried out in a more diverse set of directions and takes larger steps to converge. The reported numerical results show that the proposed method is efficient in comparison to the existing method for solving unconstrained optimization problems. However, other modified BFGS methods using the q-derivative are yet to study.

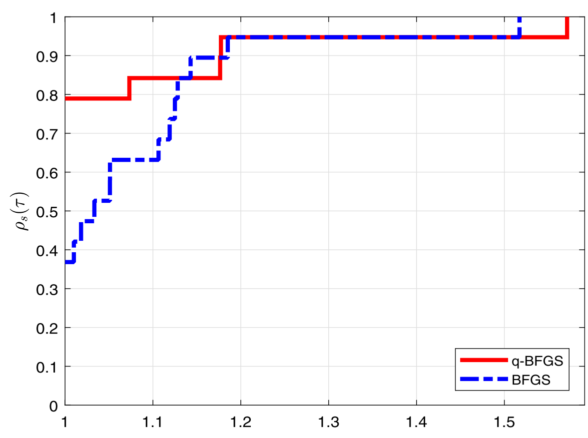
Table 7 Comparison details with [23]

Problem number	q-BFGS				
	it	*fe*	*ge*	x^*	$f(x^*)$
1	4	29	5	$(-0.062, -0.036)^T$	0.000
2	5	33	6	$(5.901, 0.509)^T$	–42.868
3	34	200	40	$(0.001, -0.012)^T$	2.36E–12
4	7	42	9	$(2.999, 0.499)^T$	9.06E–15
5	6	35	7	$(-0.002, -0.000)^T$	2.18e–06
6	8	60	9	$(0.000, 0.000)^T$	6.60e–07
7	3	21	5	$(-10.2, -6, 2.66)^T$	4.71e–18
8	5	27	6	$(9.426, 2.478)^T$	0.398
9	7	41	9	$(-0.012, 0.006)^T$	5.68e–11
10	7	39	9	$(-0.095, 0.717)^T$	–1.032
11	5	30	6	$(6.2868, 0.4000)^T$	–41.835
12	23	137	25	$(1.0, 1.0, 1.0, 1.0)^T$	2.11e–16
13	6	32	7	$(3.1417, 3.1416)^T$	–1
14	4	23	5	$(-1.9686, 16.1968)^T$	4.16e–22
15	9	73	11	$(0.0026, -1.0000)^T$	3.0006
16	7	44	8	$(0.0002, -0.0143)^T$	1.08E–07
17	9	55	10	$(4.976, 4.858)^T$	–176.542
18	12	98	14	$(0.110, 0.556, 0.852)^T$	–3.863
19	6	47	9	$(4.000, 3.999, 4.000, 3.999)^T$	–10.536

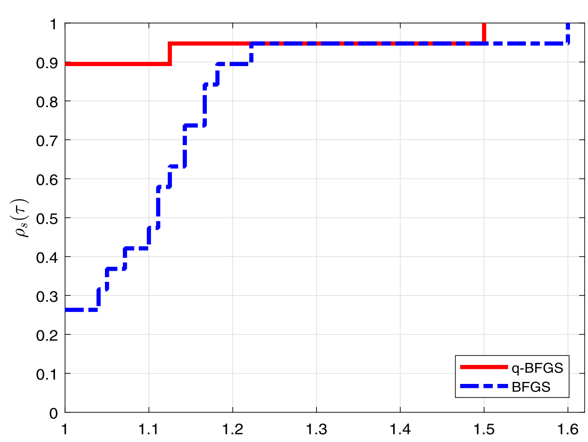
Problem number	q-BFGS				
	it	*fe*	*ge*	x^*	$f(x^*)$
1	7	44	8	$(2.6e-06, -3.3e-06)^T$	0
2	3	21	4	$(5.506, 0.507)^T$	–42.868
3	36	202	42	$(0.003, -0.016)^T$	1.3612e–11
4	8	47	10	$(2.999, 0.499)^T$	6.33e–15
5	7	40	8	$(5.23e-051.3e-06)^T$	6.46e–14
6	9	51	11	$(8.9e-06, -1.32e-05)^T$	1.39e–15
7	3	21	5	$(-10.2, -6, 2.66)^T$	4.36e–18
8	6	32	7	$(9.425, 2.475)^T$	0.398
9	7	41	9	$(-0.011, 0.006)^T$	2.66e–16
10	8	44	10	$(-0.089, 0.712)^T$	–1.032
11	5	31	7	$(6.2868, 0.4000)^T$	–41.835
12	24	144	26	$(1.0, 1.0, 1.01.0)^T$	6.12e–14
13	7	36	8	$(3.1441, 3.1422)^T$	–1
14	4	23	5	$(-1.9690, 16.1962)^T$	4.16e–22
15	10	62	13	$(0.0000, -1.0000)^T$	3
16	8	41	9	$(-6.75e-05, 5.4e-06)^T$	0
17	9	56	11	$(4.976, 4.858)^T$	–176.542
18	13	103	15	$(0.115, 0.556, 0.853)^T$	–3.863
19	6	52	8	$(4.000, 3.999, 4.000, 3.999)^T$	–10.536



(a) Performance Profiles based on number of iterations.



(b) Performance Profiles based on number of function evaluations.



(c) Performance Profiles based on number of gradient evaluations.

Figure 7 Performance profiles based on the number of iterations, the number of function evaluations and the number of gradient evaluations given in Table 7

Acknowledgements
The authors would like to thank the editors and the anonymous reviewers for their constructive comments and suggestions that have helped to improve the present paper. The fourth author was supported by Bu-Ali Sina University.

This research was supported by the Science and Engineering Research Board (Grant No. DST-SERB- MTR-2018/000121) and the University Grants Commission (IN) (Grant No. UGC-2015-UTT-59235).

Funding
Not applicable.

Availability of data and materials
Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Ethics approval and consent to participate
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Consent for publication
Not applicable.

Authors' contributions
The authors declare that the study was realized in collaboration with equal responsibility. All authors read and approved the final manuscript.

Author details
[1]Department of Mathematics, Banaras Hindu University, 221005 Varanasi, India. [2]Department of Mathematics, Indian Institute of Technology Kharagpur, Kharagpur, India. [3]Department of Mathematics, Sir Gurudas Mahavidyalaya, 700067 Kolkata, India. [4]Department of Mathematics, Bu-Ali Sina University, 65178 Hamedan, Iran. [5]DST-Centre for Interdisciplinary Mathematical Sciences, Banaras Hindu University, 221005 Varanasi, India. [6]Department of Medical Research, China Medical University Hospital, China Medical University, Taichung, Taiwan.

Publisher's Note

Received: 24 July 2020 Accepted: 3 November 2020 Published online: 12 November 2020

References

1. Mishra, S.K., Ram, B.: Introduction to Unconstrained Optimization with R. Springer, Singapore (2019). https://doi.org/10.1007/978-981-15-0894-3
2. Broyden, C.G.: The convergence of a class of double-rank minimization algorithms. IMA J. Appl. Math. **6**(1), 76–90 (1970). https://doi.org/10.1093/imamat/6.1.76
3. Fletcher, R.: A new approach to variable metric algorithms computer. Comput. J. **13**(3), 317–322 (1970). https://doi.org/10.1093/comjnl/13.3.317
4. Goldfarb, A.: A family of variable metric methods derived by variational means. Math. Comput. **24**(109), 23–26 (1970). https://doi.org/10.1090/S0025-5718-1970-0258249-6
5. Schanno, J.: Conditions of quasi-Newton methods for function minimization. Math. Comput. **24**(111), 647–650 (1970). https://doi.org/10.1090/S0025-5718-1970-0274029-X
6. Salim, M.S., Ahmed, A.R.: A quasi-Newton augmented Lagrangian algorithm for constrained optimization problems. J. Intell. Fuzzy Syst. **35**(2), 2373–2382 (2018). https://doi.org/10.3233/JIFS-17899
7. Hedayati, V., Samei, M.E.: Positive solutions of fractional differential equation with two pieces in chain interval and simultaneous Dirichlet boundary conditions. Bound. Value Probl. **2019**, 141 (2019). https://doi.org/10.1186/s13661-019-1251-8
8. Dixon, L.C.W.: Variable metric algorithms: necessary and sufficient conditions for identical behavior of nonquadratic functions. J. Optim. Theory Appl. **10**, 34–40 (1972). https://doi.org/10.1007/BF00934961
9. Samei, M.E., Yang, W.: Existence of solutions for k-dimensional system of multi-term fractional q-integro-differential equations under anti-periodic boundary conditions via quantum calculus. Math. Methods Appl. Sci. **43**(7), 4360–4382 (2020). https://doi.org/10.1002/mma.6198
10. Powell, M.J.D.: On the convergence of the variable metric algorithm. IMA J. Appl. Math. **7**(1), 21–36 (1971). https://doi.org/10.1093/imamat/7.1.21
11. Ahmadi, A., Samei, M.E.: On existence and uniqueness of solutions for a class of coupled system of three term fractional q-differential equations. J. Adv. Math. Stud. **13**(1), 69–80 (2020)
12. Dai, Y.H.: Convergence properties of the BFGS algoritm. SIAM J. Optim. **13**(3), 693–701 (2002). https://doi.org/10.1137/S1052623401383455
13. Samei, M.E., Hedayati, V., Rezapour, S.: Existence results for a fraction hybrid differential inclusion with Caputo–Hadamard type fractional derivative. Adv. Differ. Equ. **2019**, 163 (2019). https://doi.org/10.1186/s13662-019-2090-8
14. Samei, M.E., Hedayati, V., Ranjbar, G.K.: The existence of solution for k-dimensional system of Langevin Hadamard-type fractional differential inclusions with $2k$ different fractional orders. Mediterr. J. Math. **17**, 37 (2020). https://doi.org/10.1007/s00009-019-1471-2
15. Aydogan, M., Baleanu, D., Aguilar, J.F.G., Rezapour, S. Samei, M.E.: Approximate endpoint solutions for a class of fractional q-differential inclusions by computational results. Fractals **28**, 2040029 (2020). https://doi.org/10.1142/S0218348X20400290

16. Baleanu, D., Darzi, R., Agheli, B.: Fractional hybrid initial value problem featuring q-derivatives. Acta Math. Univ. Comen. **88**, 229–238 (2019)
17. Baleanu, D., Shiri, B.: Collocation methods for fractional differential equations involving non-singular kernel. Chaos Solitons Fractals **116**, 136–145 (2018). https://doi.org/10.1016/j.chaos.2018.09.020
18. Shiri, B., Baleanu, D.: System of fractional differential algebraic equations with applications. Chaos Solitons Fractals **120**, 203–212 (2019). https://doi.org/10.1016/j.chaos.2019.01.028
19. Byrd, R., Nocedal, J., Yuan, Y.: Global convergence of a class of quasi-Newton methods on convex problems. SIAM J. Numer. Anal. **24**(5), 1171–1189 (1987)
20. Byrd, R.H., Nocedal, J.: A tool for the analysis of quasi-Newton methods with application to unconstrained minimization. SIAM J. Numer. Anal. **26**(3), 727–739 (1989). https://doi.org/10.1137/0726042
21. Wei, Z., Li, G.Y., Qi, L.: New quasi-Newton methods for unconstrained optimization problems. Appl. Math. Comput. **175**(2), 1156–1188 (2006). https://doi.org/10.1016/j.amc.2005.08.027
22. Mascarenhas, W.F.: The bfgs method with exact line searches fails for non-convex objective functions. Math. Program. **99**(1), 49–61 (2004). https://doi.org/10.1007/s10107-003-0421-7
23. Li, D.H., Fukushima, M.: On the global convergence of the BFGS method for nonconvex unconstrained optimization problems. SIAM J. Optim. **11**(4), 1054–1064 (2001). https://doi.org/10.1137/S1052623499354242
24. Cieśliński, J.L.: Improved q-exponential and q-trigonometric functions. Appl. Math. Lett. **24**(12), 2110–2114 (2011). https://doi.org/10.1016/j.aml.2011.06.009
25. Ernst, T.: A method for q-calculus. J. Nonlinear Math. Phys. **10**(4), 487–525 (2003). https://doi.org/10.2991/jnmp.2003.10.4.5
26. Tariboon, J., Ntouyas, S.K.: Quantum calculus on finite intervals and applications to impulsive difference equations. Adv. Differ. Equ. **2013**, 282 (2013). https://doi.org/10.1186/1687-1847-2013-282
27. Borges, E.P.: A possible deformed algebra and calculus inspired in nonextensive thermostatistics. Phys. A, Stat. Mech. Appl. **340**(1–3), 95–101 (2004). https://doi.org/10.1016/j.physa.2004.03.082
28. Al-Saggaf, U.M., Moinuddin, M., Arif, M., Zerguine, A.: The q-least mean squares algorithm. Signal Process. **111**, 50–60 (2015)
29. Jackson, F.H.: On q-definite integrals. Q. J. Pure Appl. Math. **41**(15), 193–203 (1910)
30. Carmichael, R.D.: The general theory of linear q-difference equations. Am. J. Math. **34**, 147–168 (1912)
31. Mason, T.E.: On properties of the solution of linear q-difference equations with entire fucntion coefficients. Am. J. Math. **37**, 439–444 (1915)
32. Adams, C.R.: On the linear partial q-difference equation of general type. Trans. Am. Math. Soc. **31**, 360–371 (1929)
33. Trjitzinsky, W.J.: Analytic theory of linear q-difference equations. Acta Math. **61**, 1–38 (1933)
34. Sterroni, A.C., Galski, R.L., Ramos, F.M.: The q-gradient vector for unconstrained continuous optimization problems. In: Operations Research Proceedings 2010, pp. 365–370. Springer, Berlin (2010). https://doi.org/10.1007/978-3-642-20009-0_58
35. Diqsa, A., Khan, S., Naseem, I., Togneri, R., Bennamoun, M.: Enhanced q-least mean square. Circuits Syst. Signal Process. **38**(10), 4817–4839 (2019). https://doi.org/10.1007/s00034-019-01091-4
36. Gouvêa, E.J., Regis, R.G., Soterroni, A.C., Scarabello, M.C., Ramos, F.M.: Global optimization using the q-gradients. Eur. J. Oper. Res. **251**(3), 727–738 (2016). https://doi.org/10.1016/j.ejor.2016.01.001
37. Chakraborty, S.K., Panda, G.: q-Line search scheme for optimization problem (2017). arXiv:1702.01518
38. Chakraborty, S.K., Panda, G.: Newton like line search method using q-calculus. In: Mathematics and Computing: Third International Conference, Communications in Computer and Information Science, ICMC 2017, Haldia, India, pp. 196–208 (2017). https://doi.org/10.1007/978-981-10-4642-1_17
39. Ablinger, J., Uncu, A.K.: Functions—a mathematica package for q-series and partition theory applications (2019). arXiv:1910.12410
40. Lai, K.K., Mishra, S.K., Panda, G., Chakraborty, S.K., Samei, M.E., Ram, B.: A limited memory q-BFGS algorithm for unconstrained optimization problems. J. Appl. Math. Comput. **63**, 1–2 (2020). https://doi.org/10.1007/s12190-020-01432-6
41. Aral, A., Gupta, V., Agarwal, R.P.: Applications of q-Calculus in Operator Theory. Springer, New York (2013). https://doi.org/10.1007/978-1-4614-6946-9
42. Jackson, F.H.: On q-functions and a certain difference operator. Trans. R. Soc. Edinb. **46**, 253–281 (1909)
43. Rajković, P.M., Stanković, M.S., Marinković, S.D.: Mean value theorems in q-calculus. Mat. Vesn. **54**, 171–178 (2002)
44. Moré, J.J., Garbow, B.S., Hillstrom, K.E.: Testing unconstrained optimization software. ACM Trans. Math. Softw. **7**(1), 17–41 (1981)
45. Yuan, Y.X.: A modified bfgs algorithm for unconstrained optimization. IMA J. Numer. Anal. **11**(3), 325–332 (1991). https://doi.org/10.1093/imanum/11.3.325
46. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**(2), 201–213 (2002). https://doi.org/10.1007/s101070100263