Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 17, No. 1, pp. 57–76, 2021

A q-CONJUGATE GRADIENT ALGORITHM FOR UNCONSTRAINED OPTIMIZATION PROBLEMS*

Kin Keung Lai, Shashi Kant Mishra AND Bhagwat Ram

Abstract: We propose a q-conjugate gradient algorithm using quantum calculus for solving unconstrained optimization problems. The direction generated by the proposed method always yields a descent direction for the objective function due to q-gradient. The method does not depend on the convexity of the objective function. The parameter q is utilized in modified method with Armijo-type line search to ensure that the algorithm is a stable and fast convergence. Further, numerical results are reported to show the efficiency of the proposed method.

Key words: *conjugate gradient method, unconstrained optimization, Armijo line search, q-calculus, iterative algorithm*

Mathematics Subject Classification: *90C29, 90C53, 58E17, 05A30, 41A25*

1 Introduction

Optimization problems are presented in many applications, especially in science [20] and engineering [13, 21], image processing [16, 19], and solving M-tensor equations [22, 23], etc. The development of effective methods to solve such problems is essential. The gradient-based descent algorithms such as steepest descent [25], Newton's method [26], quasi-Newton [27] and conjugate gradient methods [28, 33] are generally fast and precise for solving large scale unconstrained optimization problems. The steepest descent and conjugate gradient descent methods are first order methods that require only the gradient of the objective function in every iteration. On the other hand, Newton and Quasi-Newton methods are second-order methods which require gradient and Hessian of the objective function in every iteration. The Newton and quasi-Newton methods require $O(n^2)$ to calculate and store Hessian or approximate Hessian matrix of n-dimension problems at each iteration. However, first-order methods only need $O(n)$ storages and calculations at each iteration. Our focus is to study the first-order method for solving problems. The conjugate gradient methods are efficient methods for solving large scale unconstrained optimization problems due to their lower computational cost and lower memory requirements.

*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).

The presence of several optima makes global optimization difficult for local optimizer methods unless the search is started to the vicinity of the global optimum or multiple starting points are supplied. For many applications where the objective function is a black-box and values are computed through a computer simulation, the gradients of the objective function are not available. Then, evolutionary algorithms [7] and simulated annealing [14] are better suited for escaping local minima for solving such complex problems but suffer from the high computational cost due to their slow convergence to the global optimum. Also, there exists no convergence proofs and the empirical convergence is quite slow. The main reason for this slow convergence is that these methods explore the global search space by creating random movements without using much local information about promising search direction [29]. In contrast, local search methods have faster convergence because of using local information to determine the most promising search direction by creating logical movements.

The quantum calculus is a novel theory that is based on finite difference re-scaling. First, q-exponential functions were discovered in q-calculus by Euler, and then Gauss, who discovered the q-binomial formula. But, the systematic development of q-calculus begins from F. H. Jackson who reintroduced the q-difference operator in 1908 for q-analogs of series, and special numbers [9]. He reintroduced the concept of the q-derivative, which is also known as the Jackson derivative [17, 18]. Recently, the use of q-derivative in the area of unconstrained optimization is studied as the q-variant of steepest descent method [11]. The results show the effective performance to escape many local minima to reach the global minimum. Searching for global optimum [11] using q-steepest descent and q-conjugate gradient methods are proposed with a stochastic approach which does not focus order of convergence of the scheme. Further, Newton and quasi-Newton methods with local and global convergent schemes using q-calculus are proposed to solve unconstrained optimization problems [4, 5] where the q-gradient vector is an extension of the classical gradient vector with the aid of the parameter q and with the property that it reduces to the classical gradient when q equals 1.

There are some well-known different conjugate gradient methods, such as the Fletcher-Reeves (FR) method [10], Hestenes-Stiefel (HS) method [15], Polak-Ribiére-Polyak (PRP) method [30, 31] and Dai-Yuan (DY) [8] method. A spectral gradient method is proposed by combining the conjugate gradient method and the spectral gradient method [3]. The reported numerical results show that the method performs well. Unfortunately, the spectral conjugate gradient method can not guarantee descent directions. Thus, the scaled conjugate algorithm [1] is developed based on quasi-Newton BFGS update formula and Wolfe line search to ensure the decrease in the objective function. Further, (FR) method is modified with different scalar parameter to converge globally [37].

In this paper, we utilize q-calculus in the modified (FR) [37] method such that the direction generated by the proposed method always provides a descent direction. We prove that the modified (FR) method with Armijo type line search due to q-gradient is globally convergent.

The paper is organized as follows. Section 2 presents q-conjugate gradient for the modified (FR) method in the context of q-calculus. Section 3 proves the global convergence of the method and the results of numerical experiments are shown in Section 4. Lastly, Section 5 provides conclusion.

2 q-Conjugate Gradient Descent for Objective Function

We first present some concepts and fundamental results related to quantum calculus. In the following, q is a positive number such that $0 < q < 1$. Let the q-integer $[n]$ be defined as $[n] = \frac{1-q^n}{1-q}$, for $n \in N$, and the derivative of x^n with respect to x be given as $[n]x^{n-1}$, then for a function $f : \mathbb{R} \to \mathbb{R}$, the q-derivative [17] is given as:

$$D_q f(x) = \frac{f(x) - f(qx)}{x - qx}, \quad x \neq 0,\ q \neq 1, \tag{2.1}$$

The q-derivative reduces to an ordinary derivative when $q \to 1$ or when $x \to 0$. The first-order partial q-derivative of function $f : \mathbb{R}^n \to \mathbb{R}$ with respect to the variable x_i, where $i = 1, \ldots, n$ is [32]:

$$D_{q_i,x_i} f(x) = \frac{f(x_1, \ldots, q_i x_i, \ldots, x_n) - f(x_1, \ldots, x_i, \ldots, x_n)}{q_i x_i - x_i}, \quad x_i \neq 0, q_i \neq 1. \tag{2.2}$$

Thus, the q-gradient is presented as the vector of n first-order partial q-derivative of f which is expressed as:

$$\nabla_q f(x)^T = \begin{bmatrix} D_{q_1,x_1} f(x) & \ldots & D_{q_i,x_i} f(x) & \ldots & D_{q_n,x_n f(x)} \end{bmatrix}, \tag{2.3}$$

where the parameter q is now a vector $q = (q_1, \ldots, q_i, \ldots, q_n)^T \in \mathbb{R}^n$. We first present the following Algorithm 1 [24, 34] to find the gradient of the function using q-calculus. However, the higher-order q-derivative of f can be found in [2].

Algorithm 1 q-Gradient Algorithm (q-GA)

1: Input $q_1 \in (0, 1), f : \mathbb{R} \to \mathbb{R}, z$.
2: **if** $x = 0$ **then**
3: Set $g \leftarrow \lim \left(\frac{f(z)-f(q*z)}{(z-q*z)}, z, 0 \right)$.
4: **else**
5: Set $g \leftarrow \frac{f(x)-f(q*x)}{(x-q*x)}$.
6: Print $\nabla_q f(x) \leftarrow g$.

Example 2.1. Let $f : \mathbb{R}^2 \to \mathbb{R}$ be defined as $f(x_1, x_2) = 2x_2^2 + 3x_1^3$. Then $\nabla_q f(x) = \begin{bmatrix} 3(1 + q + q^2)x_1^2 \\ 2(1 + q)x_2 \end{bmatrix}$.

We can also present the q-gradient for non-differentiable or discontinuous functions provided $q_i \neq 0$ and $x_i \neq 0$ for all i. The following result is due to [11] as:

Proposition 2.2. *If $f(x) = a_0 + a^T x$ where $a_0 \in \mathbb{R}$ and $a \in \mathbb{R}^n$, then for any $x, q \in \mathbb{R}^n$*

$$\nabla_q f(x) = \nabla f(x) = a. \tag{2.4}$$

Proof. The partial derivative of f with respect to x_i, where $i = 1, \ldots, n$, then

$$\frac{\partial f}{\partial x_i}(x_1, x_2, \ldots, x_i, \ldots, x_n) = \lim_{h \to \infty} \frac{f(x_1, x_2, \ldots, x_i + h, \ldots, x_n) - f(x_1, x_2, \ldots, x_i, \ldots, x_n)}{h}, \tag{2.5}$$

Let $x^T = \begin{bmatrix} x_1 & x_2 & \dots & x_i & \dots & x_n \end{bmatrix}$, and $a^T = \begin{bmatrix} a_1 & a_2 & \dots & a_i & \dots & a_n \end{bmatrix}$. Then, gradient of f is:

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(x_1, x_2, \dots, x_n) \\ \frac{\partial f}{\partial x_2}(x_1, x_2, \dots, x_n) \\ \vdots \\ \frac{\partial f}{\partial x_i}(x_1, x_2, \dots, x_n) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x_1, x_2, \dots, x_n) \end{bmatrix} = \begin{bmatrix} \lim\limits_{h\to\infty} \frac{f(x_1+h,x_2,\dots,x_i,\dots,x_n)-f(x_1,x_2,\dots,x_i,\dots,x_n)}{h} \\ \lim\limits_{h\to\infty} \frac{f(x_1,x_2+h,\dots,x_i,\dots,x_n)-f(x_1,x_2,\dots,x_i,\dots,x_n)}{h} \\ \vdots \\ \lim\limits_{h\to\infty} \frac{f(x_1,x_2,\dots,x_i+h,\dots,x_n)-f(x_1,x_2,\dots,x_i,\dots,x_n)}{h} \\ \vdots \\ \lim\limits_{h\to\infty} \frac{f(x_1,x_2,\dots,x_i,\dots,x_n+h)-f(x_1,x_2,\dots,x_i,\dots,x_n)}{h} \end{bmatrix},$$

that is,

$$\nabla f(x) = \begin{bmatrix} \lim\limits_{h\to 0} \left(\frac{a_0+a_1x_1+a_1h-a_0-a_1x_1}{h} \right) \\ \lim\limits_{h\to 0} \left(\frac{a_0+a_2x_2+a_2h-a_0-a_2x_2}{h} \right) \\ \vdots \\ \lim\limits_{h\to 0} \left(\frac{a_0+a_ix_i+a_ih-a_0-a_ix_i}{h} \right) \\ \vdots \\ \lim\limits_{h\to 0} \left(\frac{a_n+a_nx_n+a_nh-a_n-a_nx_n}{h} \right) \end{bmatrix},$$

$$\nabla f(x)^T = \begin{bmatrix} a_1 & a_2 & \dots & a_i & \dots & a_n \end{bmatrix}. \tag{2.6}$$

Using (2.2), we obtain

$$\nabla_q f(x) = \begin{bmatrix} \left(\frac{\partial f}{\partial x_1}\right)_{q_1} (x_1, x_2, \dots, x_n) \\ \left(\frac{\partial f}{\partial x_2}\right)_{q_2} (x_1, x_2, \dots, x_n) \\ \vdots \\ \left(\frac{\partial f}{\partial x_i}\right)_{q_i} (x_1, x_2, \dots, x_n) \\ \vdots \\ \left(\frac{\partial f}{\partial x_n}\right)_{q_n} (x_1, x_2, \dots, x_n) \end{bmatrix} = \begin{bmatrix} \frac{f(x_1q_1,x_2,\dots,x_i,\dots,x_n)-f(x_1,x_2,\dots,x_i,\dots,x_n)}{q_1x_1-x_1} \\ \frac{f(x_1,x_2q_2,\dots,x_i,\dots,x_n)-f(x_1,x_2,\dots,x_i,\dots,x_n)}{q_2x_2-x_2} \\ \vdots \\ \frac{f(x_1,x_2,\dots,x_iq_i,\dots,x_n)-f(x_1,x_2,\dots,x_i,\dots,x_n)}{q_ix_i-x_i} \\ \vdots \\ \frac{f(x_1,x_2,\dots,x_i,\dots,x_nq_n)-f(x_1,x_2,\dots,x_i,\dots,x_n)}{q_nx_n-x_n} \end{bmatrix},$$

that is,

$$\nabla_q f(x) = \begin{bmatrix} \frac{a_0+a_1x_1q_1-a_0-a_1x_1}{q_1x_1-x_1} \\ \frac{a_0+a_2x_2q_2-a_0-a_2x_2}{q_1x_2-x_2} \\ \vdots \\ \frac{a_0+a_ix_iq_i-a_0-a_ix_i}{q_ix_i-x_i} \\ \vdots \\ \frac{a_0+a_nx_nq_n-a_0-a_nx_n}{q_nx_n-x_n} \end{bmatrix},$$

that is,

$$\nabla_q f(x)^T = \begin{bmatrix} a_1 & a_2 & \dots & a_i & \dots & a_n \end{bmatrix}. \tag{2.7}$$

From (2.6) and (2.7), we get (2.4). □

Consider the unconstrained optimization problem:

$$\text{minimize} \quad f(x), \quad x \in \mathbb{R}^n, \tag{2.8}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is real-valued continuously q-differentiable function. We intend to utilize the conjugate gradient methods for solving (2.8). Let $x^0 \in \mathbb{R}^n$ be the starting point to solve this problem. The method generates a sequence of iterates $\{x^k\}$ recurrently through the following scheme:

$$x^{k+1} = x^k + \alpha_k d_q^k, \quad k = 0, 1, \dots, \tag{2.9}$$

where $d_q^k \in \mathbb{R}^n$ is the line search direction, and α_k is the step length moved along d_q^k [35]. For $k = 0$, the search direction is steepest descent direction and determined as:

$$d_q^0 = -\nabla_q f(x^0), \tag{2.10}$$

and for $k = 1, 2, \dots,$ we apply the following formula

$$d_q^k = -\nabla_q f(x^k) + \beta_k d_q^{k-1}, \tag{2.11}$$

where β_k is a scalar algorithmic parameter or conjugate gradient parameter. The step-length α_k is obtained by exact or inexact line searches for the global convergence of conjugate gradient methods. In the case of an exact step length, one seeks α_k along the direction d_q^k such that

$$\alpha_k = \arg\min\{f(x^k + \alpha d_q^k) | \alpha > 0\}. \tag{2.12}$$

The algorithm does not run more than n iterations to find the minimizer of quadratic functions. The parameter β_k is chosen to minimize a strictly convex function so that the direction d_q^k and d_q^{k-1} are conjugate with respect to the Hessian of the objective function. Consider an objective function in the following form:

$$f(x) = \frac{1}{2} x^T Q x - x^T b, \tag{2.13}$$

where $Q = Q^T > 0$, and $\nabla_q f(x^k) = Qx - b$.

Lemma 2.3. *For quadratic function with positive definite Hessian Q, the conjugate direction algorithm always holds in the sense of q-calculus as: $(\nabla_q f(x^{k+1}))^T d^i = 0$ for all k, $0 \le k \le n-1$, $0 \le i \le k$, and each $q_i \in (0, 1)$.*

Proof. Note that $Q(x^{k+1} - x^k) = (Qx^{k+1} - b) - (Qx^k - b)$, then

$$\nabla_q f(x^{k+1}) = \nabla_q f(x^k) + \alpha_k Q d_q^k.$$

We prove by mathematical induction. For $k = 0$:

$$(\nabla_q f(x^1))^T d_q^0 = (Qx^1 - b)^T d_q^0 = (Q(x^0 + \alpha_0 d_q^0) - b)^T d_q^0 = (Qx^0 + \alpha_0 Q d_q^0 - b)^T d_q^0,$$

that is,

$$(\nabla_q f(x^1))^T d_q^0 = (Qx^0 + \alpha_0 Q d_q^0 - b)^T d_q^0 = (x^0)^T Q d^0 + \alpha_0 (d^0)^T Q d^0 - b^T d^0.$$

For exact line search [25], we have $\alpha_0 = -\frac{(\nabla_q f(x^0))^T d_q^0}{(d_q^0)^T Q d_q^0}$, thus

$$\begin{aligned}(\nabla_q f(x^1))^T d_q^0 &= (x^0)^T Q d_q^0 - \frac{(\nabla_q f(x^0))^T d_q^0}{(d_q^0)^T Q d_q^0} (d_q^0)^T Q d_q^0 - b^T d_q^0 \\ &= (Qx^0 - b)^T d^0 - (\nabla_q f(x^0))^T d_q^0.\end{aligned}$$

Since $Qx^0 - b = \nabla_q f(x^0)$, then

$$(\nabla_q f(x^1))^T d_q^0 = (\nabla_q f(x^0))^T d_q^0 - (\nabla_q f(x^0))^T d_q^0 = 0.$$

Assume that the result is true for $k-1$, then

$$(\nabla_q f(x^k))^T d_q^i = 0 \tag{2.14}$$

for $0 \le i \le k-1$. Then, we prove that the result is true for k, that is,

$$(\nabla_q f(x^{k+1}))^T d_q^i = 0,$$

where $0 \le i \le k$. There are two cases:

1. For $0 < i < k$, we know that

$$\nabla_q f(x^{k+1}) = \nabla_q f(x^k) + \alpha_k Q d_q^k,$$

that is,

$$(\nabla_q f(x^{k+1})^T d_q^i = (\nabla_q f(x^k)^T d_q^i + \alpha_k (d_q^k)^T Q d_q^i.$$

From $Q-$conjugacy [28] in light of q-calculus, $(d_q^k)^T Q d_q^i = 0$, where $k \neq i$, and from mathematical induction hypothesis, $(\nabla_q f(x^k)^T d_q^i = 0$, thus

$$(\nabla_q f(x^{k+1}))^T d_q^i = 0,$$

for all $0 < i < k$.

2. For $i = k$,

$$\begin{aligned}(\nabla_q f(x^{k+1})^T d_q^k &= (Qx^{k+1} - b)^T d_q^k \\ &= \left(x^k - \frac{(\nabla_q f(x^k)^T d_q^k}{(d_q^k)^T Q d_q^k} d^k\right)^T Q d_q^k - b^T d_q^k \\ &= (\nabla_q f(x^k)^T d_q^k - (\nabla_q f(x^k)^T d_q^k,\end{aligned}$$

that is,

$$(\nabla_q f(x^{k+1})^T d_q^k = 0.$$

This completes the proof. $\square$

The Fletcher-Reeves (FR) method is a well known conjugate gradient method, where the parameter β_k is computed using q-gradient as

$$\beta_k = \beta_k^{FR} \equiv \frac{\|\nabla_q f(x^k)\|^2}{\|\nabla_q f(x^{k-1})\|^2}. \tag{2.15}$$

We see from (2.11) that for each $k \geq 1$, the q-derivative of f at x^k along the direction d_q^k is given by

$$\nabla_q f(x^k) d_q^k = -\|\nabla_q f(x^k)\|^2 + \beta_k^{FR} (\nabla_q f(x^k))^T d_q^{k-1}. \tag{2.16}$$

It is obvious that if exact line search is used, at that point we have $(\nabla_q f(x^k))^T d_q^{k-1} = 0$ and we obtain

$$(\nabla_q f(x^k))^T d_q^k = -\|\nabla_q f(x^k)\|^2 < 0. \tag{2.17}$$

Note that vector d_q^k is a descent direction of f at x^k. Zoutendijk proved that the (FR) method with exact line search is globally convergent [36]. The Armijo-type line search guarantees the descent property of d_q^k, that is, α_k satisfies the following inequality:

$$f(x^k + \alpha_k d_q^k) \leq f(x^k) + \delta_1 \alpha_k (\nabla_q f(x^k))^T d_q^k - \delta_2 \alpha_k^2 \|d_q^k\|^2. \tag{2.18}$$

where $\delta_1 \in (0,1)$ and $0 < \delta_2$. We present the following modification [37] due to $q-$derivative as:

$$d_q^k = -\vartheta_q^k \nabla_q f(x^k) + \beta_k^{FR} d_q^{k-1} \text{ for } k = 1, 2, \ldots, \tag{2.19}$$

where

$$\vartheta_q^k = \frac{d_q^{k-1} y_{k-1}}{\|\nabla_q f(x^{k-1})\|^2}. \tag{2.20}$$

Since $y_k = \nabla_q f(x^{k+1}) - \nabla_q f(x^k)$, then from (2.19)

$$\begin{aligned} d_q^k &= -\frac{d_q^{k-1} y_{k-1}}{\|\nabla_q f(x^{k-1})\|^2} \nabla_q f(x^k) + \beta_k^{FR} d_q^{k-1} \\ &= -\frac{d_q^{k-1} y_{k-1}}{\|\nabla_q f(x^{k-1})\|^2} \nabla_q f(x^k) + \frac{\|\nabla_q f(x^k)\|^2}{\|\nabla_q f(x^{k-1})\|^2} d_q^{k-1}, \end{aligned}$$

that is,

$$d_q^k = \frac{(d_q^{k-1})^T \nabla_q f(x^{k-1}) \nabla_q f(x^k)}{\|\nabla_q f(x^{k-1})\|^2}$$

Multiplying $\nabla_q f(x^k)$ on both sides and using (2.11), we obtain

$$(d_q^k)^T \nabla_q f(x^k) = -\|\nabla_q f(x^k)\|^2. \tag{2.21}$$

Thus, d_q^k provides a descent direction of f at x^k. Since $(\nabla_q f(x^k))^T d_q^{k-1} = 0$, then

$$\vartheta_q^k = \frac{d_q^{k-1} y_{k-1}}{\|\nabla_q f(x^{k-1})\|^2} = 1.$$

It is natural to say that the modified (FR) method reduces to the standard (FR) method as q approaches $(1,\dots,1)^T$ and $\vartheta_q^k = 1$. Based on the above discussion, we present q-Conjugate Gradient Algorithm with Armijo-Line Search (q-CGAALS) to solve unconstrained optimization problems (2.8) which is given in Algorithm 2.

Algorithm 2 q-Conjugate Gradient Algorithm with Armijo-Line Search (q-CGAALS)

Given constants $\delta_1 \in (0,1), \rho \in (0,1), \delta_2 > 0$, $q_i^0 \in (0,1)$ for $i = 0,\dots,n-1$. Choose a starting point $x^0 \in \mathbb{R}^n$.
Compute $\nabla_q f(x^0)$ using Algorithm 1.
if $\|\nabla f(x^0)\| \le \epsilon$ **then**
    Stop.
else
    Set $d_q^0 \leftarrow -\nabla_q f(x^0)$.
for k=0,1,2,... **do**
    Determine a stepsize $\alpha_k = \max\{\rho^{-j}, j = 0,1,2,\dots\}$ satisfying

$$f(x^k + \alpha_k d_q^k) \le f(x^k) + \delta_1 \alpha_k (\nabla_q f(x^k))^T d_q^k - \delta_2 \alpha_k^2 \|d_q^k\|^2. \tag{2.22}$$

    Update $x^{k+1} \leftarrow x^k + \alpha_k d_q^k$.
    if $\|\nabla f(x^{k+1})\| \le \epsilon$ **then**
        Stop.
    else
        Compute $\beta_k \leftarrow \beta_k^{FR}$ using (2.15).
    Compute the search direction d_q^k by (2.19).

Remark 2.4. 1. The starting vector parameter $(q_0,\dots,q_{n-1})$ is chosen in such a way that each $q_i \in (0,1)$ for $i = 0,\dots,n-1$. Further, a suitable mechanism is attained to generate the next q for each $0 < q_i < 1$, where $i = 0,\dots n-1$. The detailed description about this mechanism is given in Section 3.

2. The stopping criteria is given as the general gradient of objective function which is computed at each iterative point x^k.

Note that d_q^k is a descent direction in the context of q-derivative. If vector q does not approach to $(1,\dots,1)^T$, then search directions are not necessarily descent directions and this makes it possible for Algorithm 2 to escape from local minima to global minima. Our direction generated by modified Fletcher-Reeves given in [37] is modified by replacing the gradient with q-gradient which always possess a descent direction due to (2.19), and does not depend on any line search used, so the modified Armijo line search with backtracking is utilized under mild conditions to obtain the global convergence of Algorithm 2.

3 Global Convergence

In this section, we need the following assumptions on the objective function to prove the global convergence.

Assumption 3.1. 1. Let x^0 be a starting point for the iteration (2.9), (2.10) and (2.11) such that the level set $\Omega := \{x \in \mathbb{R}^n | f(x) \leq f(x^0)\}$ is bounded.

2. In some neighborhood N of Ω, f is continuously q-differentiable and for $L > 0$, we have

$$\|\nabla_q f(x) - \nabla_q f(y)\| \leq L\|x - y\|, \quad \forall\, x, y \in N. \tag{3.1}$$

From Proposition 2.2, condition (3.1) of Assumption 3.1 is obvious for classical gradient. Thus, q-gradient and classical gradient provide same result as we have proved. To show that condition (3.1) of Assumption 3.1 holds under general gradient for satisfying the Lipschitz condition, we compute each q_i [4] as: With a starting number $q_0^0 \in (0, 1)$, for $k = 0, 1, \ldots$

$$q_i^{k+1} = 1 - \frac{q_i^k}{(k+1)^2}, \tag{3.2}$$

where $i = 0, \ldots, n-1$. It is worth mentioning that each q_i finally approaches 1 when $k \to \infty$. We present the following example to show the numerical solution of q-gradient and general gradient for a given function.

Example 3.2. Consider a function $f : \mathbb{R}^3 \to \mathbb{R}$ such that $f(x) = 2x_1^2 - x_2^2 + 3x_3^3 + 5$.

We find the q-gradient by taking $(q_0^0, q_1^0, q_2^0)^T = (0.91, 0.91, 0.91)^T$, we run Algorithm 1 for $k = 30$ iterations. The generated numerical values are provided in Table 1 where the last column is the computed q-gradient of the function at x. At $x = (1, -1, 1)^T$, and $q = (q_0^{29}, q_1^{29}, q_2^{29}) = (0.998812\ ,\ 0.998812\ ,\ 0.998812)^T$, we get the q-gradient of f as:

$$\nabla_q f(x) = (3.997625\ ,\ 1.998812\ ,\ 8.989316)^T.$$

On the other hand, general gradient of f provides:

$$\nabla f(x) = (4.0000\ ,\ 2.0000\ ,\ 9.0000)^T.$$

The three dimension graphics is given in Figure 1.

Remark 3.3. 1. From Proposition 2.2, both q-gradient and gradient vectors are same.

2. For other nonlinear functions, we have provided Example 3.2 which demonstrates that the q-gradient and gradient vectors are almost same for large value of k as q_i, where $i = 1, \ldots, n$, approaches 1 due to (3.2). From computation point of view, we obtain approximation results for both vectors.

Thus, from the above two justifications, condition (3.1) of Assumption 3.1 also holds for the general gradient and subsequently satisfies for the general gradient Lipschitz condition.

The sequence $\{x^k\}$ generated by Algorithm 2 is in Ω. Thus, the sequence $\{f(x^k)\}$ is also decreasing. For constant $\gamma > 0$, we have

$$\|\nabla_q f(x^k)\| \leq \gamma, \quad \forall\, x \in \Omega. \tag{3.3}$$

Table 1: q-Gradient Iteration using (3.2) for Example 3.2.

k	$q = (q_0^k\ ,\ q_1^k\ ,\ q_2^k)^T$	$f(x^k)$	$f(qx^k)$	$\nabla_q(x^k)$
0	$(0.910000\ ,\ 0.910000\ ,\ 0.910000)^T$	9	8.260713	$(3.820000\ ,\ 1.910000\ ,\ 8.214300)^T$
1	$(0.090000\ ,\ 0.090000\ ,\ 0.090000)^T$	9	6.002187	$(2.180000\ ,\ 1.090000\ ,\ 3.294300)^T$
2	$(0.977500\ ,\ 0.977500\ ,\ 0.977500)^T$	9	8.802022	$(3.955000\ ,\ 1.977500\ ,\ 8.799019)^T$
3	$(0.891389\ ,\ 0.891389\ ,\ 0.891389)^T$	9	8.124824	$(3.782778\ ,\ 1.891389\ ,\ 8.057889)^T$
4	$(0.944288\ ,\ 0.944288\ ,\ 0.944288)^T$	9	8.526009	$(3.888576\ ,\ 1.944288\ ,\ 8.507905)^T$
5	$(0.962228\ ,\ 0.962228\ ,\ 0.962228)^T$	9	8.672735	$(3.924457\ ,\ 1.962228\ ,\ 8.664336)^T$
6	$(0.973271\ ,\ 0.973271\ ,\ 0.973271)^T$	9	8.765815	$(3.946543\ ,\ 1.973271\ ,\ 8.761586)^T$
7	$(0.980137\ ,\ 0.980137\ ,\ 0.980137)^T$	9	8.824763	$(3.960275\ ,\ 1.980137\ ,\ 8.822419)^T$
8	$(0.984685\ ,\ 0.984685\ ,\ 0.984685)^T$	9	8.864268	$(3.969371\ ,\ 1.984685\ ,\ 8.862872)^T$
9	$(0.987843\ ,\ 0.987843\ ,\ 0.987843)^T$	9	8.891915	$(3.975687\ ,\ 1.987843\ ,\ 8.891034)^T$
10	$(0.990122\ ,\ 0.990122\ ,\ 0.990122)^T$	9	8.911969	$(3.980243\ ,\ 1.990122\ ,\ 8.911387)^T$
11	$(0.991817\ ,\ 0.991817\ ,\ 0.991817)^T$	9	8.926956	$(3.983634\ ,\ 1.991817\ ,\ 8.926555)^T$
12	$(0.993112\ ,\ 0.993112\ ,\ 0.993112)^T$	9	8.938437	$(3.986225\ ,\ 1.993112\ ,\ 8.938154)^T$
13	$(0.994124\ ,\ 0.994124\ ,\ 0.994124)^T$	9	8.947423	$(3.988247\ ,\ 1.994124\ ,\ 8.947216)^T$
14	$(0.994928\ ,\ 0.994928\ ,\ 0.994928)^T$	9	8.954583	$(3.989856\ ,\ 1.994928\ ,\ 8.954429)^T$
15	$(0.995578\ ,\ 0.995578\ ,\ 0.995578)^T$	9	8.960379	$(3.991156\ ,\ 1.995578\ ,\ 8.960262)^T$
16	$(0.996111\ ,\ 0.996111\ ,\ 0.996111)^T$	9	8.965135	$(3.992222\ ,\ 1.996111\ ,\ 8.965045)^T$
17	$(0.996553\ ,\ 0.996553\ ,\ 0.996553)^T$	9	8.969086	$(3.993106\ ,\ 1.996553\ ,\ 8.969015)^T$
18	$(0.996924\ ,\ 0.996924\ ,\ 0.996924)^T$	9	8.972403	$(3.993848\ ,\ 1.996924\ ,\ 8.972346)^T$
19	$(0.997238\ ,\ 0.997238\ ,\ 0.997238)^T$	9	8.975215	$(3.994477\ ,\ 1.997238\ ,\ 8.975169)^T$
20	$(0.997507\ ,\ 0.997507\ ,\ 0.997507)^T$	9	8.977618	$(3.995014\ ,\ 1.997507\ ,\ 8.977581)^T$
21	$(0.997738\ ,\ 0.997738\ ,\ 0.997738)^T$	9	8.979689	$(3.995476\ ,\ 1.997738\ ,\ 8.979658)^T$
22	$(0.997939\ ,\ 0.997939\ ,\ 0.997939)^T$	9	8.981485	$(3.995877\ ,\ 1.997939\ ,\ 8.98146)^T$
23	$(0.998114\ ,\ 0.998114\ ,\ 0.998114)^T$	9	8.983054	$(3.996227\ ,\ 1.998114\ ,\ 8.983033)^T$
24	$(0.998267\ ,\ 0.998267\ ,\ 0.998267)^T$	9	8.984431	$(3.996534\ ,\ 1.998267\ ,\ 8.984413)^T$
25	$(0.998403\ ,\ 0.998403\ ,\ 0.998403)^T$	9	8.985648	$(3.996806\ ,\ 1.998403\ ,\ 8.985633)^T$
26	$(0.998523\ ,\ 0.998523\ ,\ 0.998523)^T$	9	8.986727	$(3.997046\ ,\ 1.998523\ ,\ 8.986714)^T$
27	$(0.998630\ ,\ 0.998630\ ,\ 0.998630)^T$	9	8.987689	$(3.997261\ ,\ 1.99863\ ,\ 8.987678)^T$
28	$(0.998726\ ,\ 0.998726\ ,\ 0.998726)^T$	9	8.988551	$(3.997452\ ,\ 1.998726\ ,\ 8.988541)^T$
29	$(0.998812\ ,\ 0.998812\ ,\ 0.998812)^T$	9	8.989325	$(3.997625\ ,\ 1.998812\ ,\ 8.989316)^T$

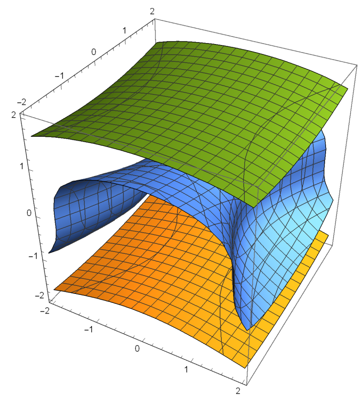

Figure 1: 3D Graphics of Example 3.2

Theorem 3.4. *Let $\{x^k\}$ and $\{d_q^k\}$ be generated by Algorithm 2 and there exists a constant $c_1 > 0$ such that the following inequality holds for all k sufficiently large,*

$$\alpha_k \geq c_1 \frac{\|\nabla_q f(x^k)\|^2}{\|d_q^k\|^2}. \tag{3.4}$$

Proof. From the given Assumption 3.1 and (2.22), we have

$$\sum_{k=0}^{\infty} (-\delta_1 \alpha_k (\nabla_q f(x^k))^T d_q^k + \delta_2 \alpha_k^2 \|d_q^k\|^2) < \infty. \tag{3.5}$$

This together with (2.21) yields

$$\sum_{k \geq 0} \alpha_k^2 \|d_q^k\|^2 < \infty, \tag{3.6}$$

and

$$\sum_{k \geq 0} \alpha_k \|\nabla_q f(x^k)\|^2 = -\sum_{k \geq 0} \alpha_k (\nabla_q f(x^k))^T d_q^k < \infty.$$

In particular, we have

$$\lim_{k \to \infty} \alpha_k \|d_q^k\| = 0,$$

and

$$\lim_{k \to} \alpha_k \|\nabla_q f(x^k)\| = 0. \tag{3.7}$$

We now prove (3.4) by considering the following two cases:

1. $\alpha_k = 1$,

2. $\alpha_k < 1$.

When we take Case 1, then we get $\|\nabla_q f(x^k)\| \leq \|d_q^k\|$. In this case, inequality (3.4) is satisfied with $c_1 = 1$. From the mean value theorem in q-calculus, there is a $t^k \in (0,1)$ such that $x^k + t^k \rho^{-1}\alpha_k d_q^k \in \mathbb{N}$ and

$$\begin{aligned} f(x^k + \rho^{-1}\alpha_k d_q^k) - f(x^k) &= \rho^{-1}\alpha_k \nabla_q f(x^k + t^k\rho^{-1}\alpha_k d_q^k)^T d_q^k \\ &= \rho^{-1}\alpha_k \nabla_q f(x^k) d_q^k + \rho^{-1}\alpha_k(\nabla_q f(x^k + \alpha_k d_q^k)^T) d_q^k \\ &\leq \rho^{-1}\alpha_k \nabla_q f(x^k)^T d_q^k + L\rho^{-2}\alpha_k^2 \|d_q^k\|^2. \end{aligned}$$

Substituting the last inequality into (3.3), we get

$$\alpha_k > \frac{(1-\delta_1)\rho\|\nabla_q f(x^k)\|^2}{(L+\delta_2)\|d_q^k\|^2}. \tag{3.8}$$

□

The Armijo line search is an important inexact line search, and it is very simple because it requires only one gradient evaluation per iteration. In practice a line search procedure may have to be equipped with several mechanism that guarantee that a step-size satisfying the termination criteria will indeed be obtained. We have used q-derivative to compute the q-gradient and replaced general gradient. We solve several numerical test problem to show the advantage of q-gradient in backtracking Armijo line search for obtaining the least number of iterations and function evaluations, respectively.

Feasibility of Backtracking Armijo Line Search with q-gradient: Note that (2.22) establishes convergence to stationary points of smooth functions using an inexact line search with a simple sufficient decrease condition. Armijo condition ensures that the line search step is not too large. To prevent large steps relative to decreasing of f, we require (2.22) with $\delta_1 \in (0,1)$. Typical values of δ_1 ranges from 10^{-4} to 0.1. As q approaches $(1,\ldots,1)^T$ for large value of k, backtracking Armijo search in the sense of q-calculus starts to behave as a classical backtracking Armijo search. Moreover, due to inclusion of q-gradient, the value of step length is responsible to converge fast in comparison to the classical condition. Note that x^k is generated by Algorithm 1 with backtracking Armijo line-search, then we find a q-stationary point in a finite number of steps, which is eventually an approximation of general stationary point, but if the function f is unbounded below , so the minimum does not exist. If $\nabla_q(x^k)$ and d_q^k do not become orthogonal and $\|d_q^k\| \nrightarrow 0$, then $\|\nabla_q f(x)\| \to 0$ when q approaches $(1,\ldots,1)^T$ as $k \to \infty$. Thus, this is necessary to prove the global convergence theorem in the q-calculus context.

Theorem 3.5. *Suppose $\{x^k\}$ is generated by Algorithm 2. Then, for some k, we have*

$$\lim_{k\to\infty} \inf \|\nabla_q f(x^k)\| = 0.$$

Proof. From the sake of contradiction, we suppose that the conclusion is not true. Then, there exists a constant $\epsilon > 0$ such that

$$\|\nabla_q f(x^k)\| \geq \epsilon, \quad k \geq 0. \tag{3.9}$$

We get

$$\|d_q^k\|^2 = (\beta_k^{FR})^2\|d_q^{k-1}\|^2 - 2\vartheta_q^k (d_q^k)^k \nabla_q f(x^k) - (\vartheta_q^k)^2 \|\nabla_q f(x^k)\|^2.$$

Dividing both sides of this inequality by $(\nabla_q f(x^k)^T d_q^k)$, we get

$$\begin{aligned}
\frac{\|d_q^k\|^2}{\|\nabla_q f(x^k)\|^4} &= \frac{\|d_q^k\|}{((\nabla_q f(x^k))^T d_q^k)^2} = (\beta_k)^2 \frac{\|d_q^{k-1}\|^2}{(\nabla_q f(x^k) d_q^k)^2} - \frac{2\vartheta_q^k}{(d_q^k)^T \nabla_q f(x^k)} - \frac{2\vartheta_q^k}{d_q^k(\nabla_q f(x^k))}. \\
&= \left(\frac{\|d_q^k\|^2}{(\nabla_q f(x^{k-1}))^T}\right)^2 \frac{\|d_q^{k-1}\|^2}{\|\nabla_q f(x^k)\|^4} + \frac{2\vartheta_q^k}{\|\nabla_q f(x^k)\|^2} - \frac{\vartheta_q^k}{|\nabla_q f(x^k)\|^2} \\
&= \frac{\|d_q^{k-1}\|^2}{\|\nabla_q f(x^{k-1})\|^4} - \frac{(\vartheta_q^k - 1)^2}{\|\nabla_q f(x^{k-1})\|^2} + \frac{1}{\|\nabla_q f(x^{k-1})\|^2} \\
&\leq \frac{\|d_q^{k-1}\|^2}{\|\nabla_q f(x^{k-1})\|^4} + \frac{1}{\|\nabla_q f(x^{k-1})\|^2} \\
&\leq \sum_{j=0}^{k-1} \frac{1}{\|\nabla_q f(x^j)\|^2}.
\end{aligned}$$

The last inequalities implies

$$\sum_{k\geq 1} \frac{\|\nabla_q f(x^k)\|}{d_q^k} \geq \epsilon^2 \sum_{k\geq 1} \frac{1}{k} = \infty.$$

This is a contradiction. Thus, the proof is complete. □

Proposition 3.6. *Let $\{x^k\}$ be the sequence generated by a q-gradient method $x^{k+1} = x^k + \alpha_k d_q^k$ and satisfies $\nabla_q f(x^k) \to 0.$, then every limit point of sequences that it generates is a q-stationary point of f.*

Proof. There exists a subsequence $\{x^k\}_K$ and $\epsilon > 0$ such that $|\nabla_q f(x^k)| \geq \epsilon$ for all $k \in K$. Since $\{x^k\}_K$ is bounded it has at least one limit point x^* and we must have $|\nabla_q f(x^*)| \geq \epsilon$ But this contradicts our hypothesis, which implies that x^* must be a q-stationary point. □

4 Numerical Results

We now report the numerical performance of Algorithm 2. We compare our numerical performance with modified Fletcher-Reeves (CG(MFR)) method given in [37] and report that our method is very efficient to solve non-convex unconstrained optimization problems. The parameters in Algorithm 2 is specified by

$$\epsilon = 10^{-6} \quad \rho = \frac{1}{2}, \quad \delta_1 = 10^{-3}, \quad \delta_2 = 10^{-8}.$$

All codes (q-CGAALS) and ($CG(MFR)$) are written in MATLAB (2017a) and run on a personal laptop equipped with Intel(R) Core(TM) i3-4005U CPU, 1.70 GHz CPU processor, 4 GB RAM memory, and Windows 10 operating system. We have taken general gradient norm

$$\nabla f(x^k) \leq \epsilon$$

for both algorithms as stopping criteria to terminate. We solve test problems using 50 different starting points and report the part of numerical results in Table 2, Table 3, and Table 4. The following notations are used in tables:

IT : Number of iterations

Table 2: Numerical Summary for Example 4.1

Sl.No.	$(q_0\ ,\ q_1)^T$	Starting Point	x^*	$f(x^*)$	Fe	IT
1	$(0.9758\ ,\ 0.9758)^T$	$(-3.9613\ ,\ -3.4445)^T$	$(-2.9971\ ,\ -3.006)^T$	-78.1607	57	4
2	$(0.9651\ ,\ 0.9651)^T$	$(-3.4938\ ,\ -0.3831)^T$	$(-2.9092\ ,\ -3.0132)^T$	-78.3314	37	6
3	$(0.9825\ ,\ 0.9825)^T$	$(-2.6454\ ,\ -2.849)^T$	$(-2.9006\ ,\ -2.9292)^T$	-78.3209	11	2
4	$(0.9713\ ,\ 0.9713)^T$	$(-3.8476\ ,\ -4.0759)^T$	$(-2.9317\ ,\ -2.9639)^T$	-78.308	26	4
5	$(0.9640\ ,\ 0.9640)^T$	$(-0.7785\ ,\ -0.4756)^T$	$(-2.9974\ ,\ -2.9616)^T$	78.1349	50	6
6	$(0.9876\ ,\ 0.9876)^T$	$(-4.0262\ ,\ -0.1013)^T$	$(-2.9081\ ,\ -2.9311)^T$	-78.3306	38	6
7	$(0.9850\ ,\ 0.9850)^T$	$(-3.8704\ ,\ -2.8057)^T$	$(-2.9083\ ,\ -2.9466)^T$	-78.3319	39	4
8	$(0.9720\ ,\ 0.9720)^T$	$(-4.1465\ ,\ -3.4444)^T$	$(-3.0743\ ,\ -3.0321)^T$	-77.7651	35	8
9	$(0.9749\ ,\ 0.9749)^T$	$(-3.8617\ ,\ -3.7097)^T$	$(-2.9495\ ,\ -2.9935)^T$	-78.2914	14	2
10	$(0.9735\ ,\ 0.9735)^T$	$(-2.8215\ ,\ -2.9564)^T$	$(-2.9130\ ,\ -2.9764)^T$	-78.3301	36	3

Table 3: Comparative results for different q for Example 4.2

Sl. No.	$(q_0\ \ q_1)^T$	Starting Point	$q-CGAALS$				$CG(MFR)$		IT	F_e
			x^*	$f(x^*)$	IT	F_e	x^*	$f(x^*)$		
1	$(0.9696\ ,\ 0.9762)^T$	$(1.2363\ ,\ -1.5076)^T$	$(3.4359\ ,\ -2.3260)^T$	1.84851	3	16	$(3.3972\ ,\ -2.2679)^T$	1.78E+00	6	19
2	$(0.9636\ ,\ 0.9636)^T$	$(2.8090\ ,\ -1.4694)^T$	$(3.4499\ ,\ -2.3196)^T$	1.94E+00	3	14	$(3.3972\ ,\ -2.2679)^T$	1.78E+00	7	18
3	$(0.9861\ ,\ 0.9861)^T$	$(1.3385\ ,\ -1.0357)^T$	$(3.4024\ ,\ -2.2994)^T$	1.78E+00	5	23	$(3.3972\ ,\ -2.2679)^T$	1.78E+00	5	23
4	$(0.9582\ ,\ 0.9582)^T$	$(2.7230\ ,\ -1.1945)^T$	$(3.4257\ ,\ -2.4191)^T$	1.86E+00	4	34	$(3.3972\ ,\ -2.2679)^T$	1.78E+00	8	23
5	$(0.9832\ ,\ 0.9832)^T$	$(2.4172\ ,\ -2.3454)^T$	$(3.3972\ ,\ -2.3056)^T$	1.78E+00	4	43	$(3.3972\ ,\ -2.2679)^T$	1.78E+00	6	16
6	$(0.9579\ ,\ 0.9579)^T$	$(1.9407\ ,\ -2.7557)^T$	$(3.4182\ ,\ -2.3691)^T$	1.80E+00	4	23	$(3.3972\ ,\ -2.2679)^T$	1.78E+00	8	35
7	$(0.9672\ ,0.9672)^T$	$(1.1690\ ,\ -1.2833)^T$	$(3.4450\ ,\ -2.3181)^T$	1.90E+00	5	27	$(3.4450, -2.3181)^T$	1.78E+00	5	27
8	$(0.9801\ ,\ 0.9801)^T$	$(1.0974\ ,\ -2.1683)^T$	$(3.4086\ ,\ -2.3151)^T$	1.78E+00	2	17	$(3.3972\ ,\ -2.2679)^T$	1.78E+00	5	17
9	$(0.9879\ ,\ 0.9879)^T$	$(2.7013\ ,\ -2.1042)^T$	$(3.3978\ ,\ -2.3059)^T$	1.78E+00	3	19	$(3.3972\ ,\ -2.2679)^T$	1.78E+00	5	12
10	$(0.9528\ ,\ 0.9528)^T$	$(2.7933\ ,-1.4550)^T$	$(3.4600\ ,\ -2.3351)^T$	2.01E+00	3	14	$(3.3972\ ,\ -2.2679)^T$	1.78E+00	7	15
11	$(0.9954\ ,\ 0.9954)^T$	$(0.3675\ ,\ -2.0443)^T$	$(3.4011\ ,\ -2.2740)^T$	1.78E+00	6	25	$(3.3972\ ,\ -2.2679)^T$	1.78E+00	8	23

F_e : Number of function evaluations

x^* : Minimizer

$f(x^*)$: Minimum function value

Example 4.1. Consider a test function $f : \mathbb{R}^3 \to \mathbb{R}$ such that $f(x) = \frac{1}{2}\big(x_1^4 + x_2^4 - 16x_1^2 - 16x_2^2 + 5x_1 + 5x_2\big)$.

The function is continuous and non-convex and multimodal function. The search space is $[-5, 5]$ and we choose ten different starting points with ten different values of vector q. On MATLAB platform, with these starting points, and tolerance limit of the general gradient norm 10^{-6}, the proposed Algorithm 2 reaches to the solution point. Results are summarized in Table 2 for 10 different q as follows:

Example 4.2. Consider a function $f : \mathbb{R}^2 \to \mathbb{R}$ such that $f(x) = (x_1^2 + x_2 - 10)^2 + (x_1 + x_2^2 - 7)^2 + (x_1^2 + x_2^3 - 1)^2$.

This is Continuous, Differentiable, Non-Scalable, and multimodal function. The global minimization is located at $x^* = (3.4091\ ,\ -2.1714)^T$ and $f(x^*) = 1.7127$. Starting with the initial 11 random points generated from the search space $[-3, 3]$, we are looking the solution point for 11 different vector q given in Table 3. We conclude that the proposed algorithm converges fast in comparison to (CG(MFR)) with the least number of iterations and function evaluations. The graphics of the function is given in Figure 2.

We now report some numerical experiments as given in Table 4. We test Algorithm 2 on well-known 31 test problems from CUTE library [12] where 15 test functions are non-convex,

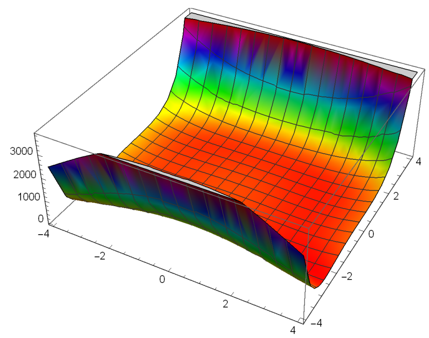

Figure 2: 3D Graphics of Example 4.2

Table 4: Summary of 31 Numerical Experiments

Description of Test Functions					q-CGAALS		CG(MFR)	
Sl. No.	Test Function	Search Domain	Nature	Starting Point	IT	F_e	IT	F_e
1	Ackley Function	[-15, 30]	Convex	$(0.4, 0.3)^T$	4	13	15	68
2	Beale	$[-4.5\ ,\ 4.5]$	Non-Convex	$(1\ ,\ 2)^T$	10	23	13	57
3	Bohachevsky	$[-100\ ,\ 100]$	Convex	$(5\ ,\ 3)^T$	4	14	6	26
4	Booth	$[-10\ ,\ 10]$	Convex	$(6\ ,\ -1)^T$	4	9	5	14
5	Branin	$[-5\ ,10]\ [0\ ,15]$	Non-Convex	$(-3\ ,\ 0)^T$	7	22	7	17
6	Brent	[-10, -10]	Convex	$(-5, -5)^T$	3	5	4	28
7	Camel 3 Hump	[-5, 5]	Non-Convex	$(-1, -5)^T$	8	17	9	17
8	Dixon & Price	$[-10\ ,\ 10]$	Non-Convex	$(-3\ ,\ 1)^T$	8	20	9	25
9	Extended Beale	$[-4.5\ ,\ 4.5]$	Non-Convex	$(1, 08, 1, 0.8)^T$	8	19	9	27
10	Freudenstein-Rooth	[-10 , 10]	Non-Convex	$(0.5, 0.5, 0.5, 0.5)^T$	8	25	9	37
11	Goldstein & Price	$[-2\ ,\ 2]$	Non-Convex	$(1\ ,\ 1)^T$	7	38	11	50
12	Griewank	$[-600\ ,\ 600]$	Non-Convex	$(1\ ,\ 3)^T$	8	21	9	20
13	Himmelblau	[-6 , 6]	Non-Convex	$(1.5, 1.5, 0.5, 1.5)^T$	14	30	14	37
14	Humps	$[-5\ ,\ 5]$	Non-Convex	$(-4\ ,\ 4)^T$	9	25	9	30
15	Hyper-Ellipsoid	$[-65.53\ ,\ 65.53]$	Convex	$(1\ ,\ 33)^T$	4	10	3	8
16	Levy	$[-10\ ,\ 10]$	Non-Convex	$(4\ ,\ 6)^T$	5	15	8	21
17	Matyas	$[-10\ ,\ 10]$	Convex	$(-3\ ,\ -1)^T$	5	14	3	7
18	McCormick	$[-1.5, 4]\ [-3, 4]$	Convex	$(1\ ,\ -2)^T$	4	9	5	18
19	Michalewicz	$[0\ ,\ \pi]$	Non-Convex	$(2.1\ ,\ 3.5)^T$	9	55	10	22
20	Prem	$[-1\ ,\ 2]$	Convex	$(1\ ,\ 1.5)^T$	6	16	7	20
21	Perturbed Quadratic	$[-4\ ,\ 5]$	Convex	$(1\ ,\ -2)^T$	4	7	4	17
22	Power	$[0\ ,\ 4]$	Convex	$(1\ ,\ 2\ ,\ 1\ ,\ 1)^T$	10	39	14	74
23	Rastrigin	$[-5.12\ ,\ 5.12]$	Non-Convex	$(-4.1\ ,\ 1.7)^T$	5	13	6	22
24	Rosenbrock	$[-5\ ,\ 10]$	Non-Convex	$(-3\ ,\ 2)^T$	17	37	19	47
25	Schwelel	$[-500\ ,\ 500]$	Convex	$(1\ ,\ 2)^T$	4	27	6	17
26	Schaffer	$[-100\ ,\ 100]$	Non-Convex	$(-3\ ,\ 1)^T$	20	12	22	54
27	Sphere	[0, 10]	Convex	$(-1, 2.3)^T$	3	8	2	5
28	Shekel	[0, 10]	Convex	$(4, 3, 2, 1)^T$	9	41	8	44
29	Sum Squares	$[-5.12\ ,\ 5.12]$	Convex	$(-1.65\ ,\ 4.76)^T$	3	7	4	17
30	Trid	$[-4\ ,\ 4]$	Convex	$(1\ ,\ 4)^T$	5	10	6	20
31	Zakharov	[-5, 10]	Convex	$(-1, 3)^T$	4	11	6	26

and compare its performance with the methodology used in [37], that is, (CG(MFR)). We compare the performance of (q-CGAALS) with ($CG(MFR)$) using the performance profiles introduced in [6], which is suitable when function evaluations F_e constitute the dominant

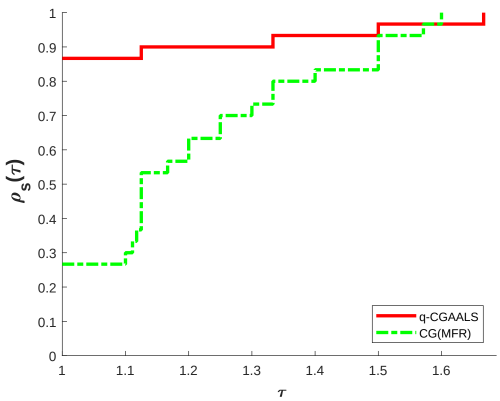

Figure 3: Performance Profile for Number of Iterations

computational cost while running Algorithm 2. The performance profile in Figure 3 shows that (q-CGAALS) is competitive than the (CG(MFR)) in term of number of iterations IT. The comparison graph in terms of number of function evaluations is also given in Figure 4.

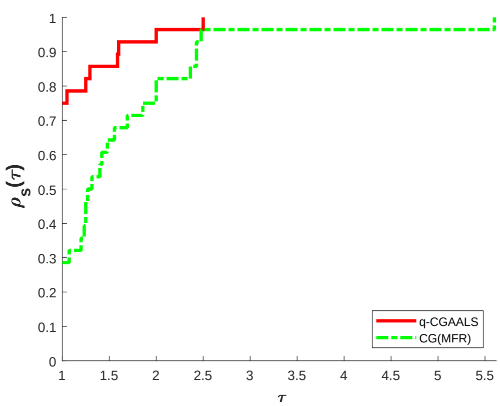

Figure 4: Performance Profile for Number of Function Evaluations

5 Conclusions

In this paper, quantum calculus is used in the Armijo type line search to decrease the value of objective function. The global convergence of the proposed algorithm has been provided under mild conditions. In numerical experiments, the search process gradually moves from global search in the beginning to the local search in the end and it is shown that the proposed method is promising. From applications point of view, the authors hope that this concept may be further extended for multiobjective optimization problems.

Acknowledgment

The authors are grateful to the anonymous referees for their valuable comments and suggestions on the paper which have greatly improved the paper.

References

[1] N. Andrei, Scaled conjugate gradient algorithms for unconstrained optimization, *Computational Optimization and Applications* 38 (2007) 401–416.

[2] C.R. Adams, The general theory of a class of linear partial difference equations, *Transactions of the American mathematical society* 26 (1924) 183–312.

[3] E. G. Birgin and J. M. Martinez, A spectral conjugate gradient method for unconstrained optimization, *Applied Mathematics and optimization* 43 (2001) 117–128.

[4] S.K. Chakraborty and G. Panda, Newton like line search method using q-calculus in: *Mathematics and Computing. ICMC 2017. Communications in Computer and Information Science*, D. Giri, R. Mohapatra, H. Begehr and M. Obaidat (Eds.), Springer, Singapore, 2017, pp. 196–208.

[5] S.K. Mishra, G. Panda, M.A.T. Ansary and B. Ram, On q-Newton's method for unconstrained multiobjective optimization problems, *Journal of Applied Mathematics and Computing* 63 (2020) 391–410.

[6] E.D. Dolan and J.J. Moré, Benchmarking optimization software with performance profiles, *Mathematical programming* 91 (2002) 201–213.

[7] K. Deb and S. Tiwari, Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization, *European Journal of Operational Research* 185 (2008) 1062–1087.

[8] Y.H. Dai and Y. Yuan, A nonlinear conjugate gradient method with a strong global convergence property, *SIAM Journal on optimization* 10 (1999) 177–182.

[9] T. Ernst, A method for q-calculus, *Journal of Nonlinear Mathematical Physics* 10 (2003) 487–525.

[10] R. Fletcher, *Practical Methods of Optimization*, 2nd Ed., John Wiley & Sons, New York, USA, 1987.

[11] É. J. Gouvêa, R.G. Regis, A.C. Soterroni, M.C. Scarabello and F.M. Ramos, Global optimization using q-gradients, *European Journal of Operational Research* 251 (2016) 727–738.

[12] N. I. Gould, D. Orban and P. L. Toint, CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization, *Computational Optimization and Applications* 60 (2015) 545–557.

[13] S. Hubmer, A. Neubauer, R. Ramlau and H.U. Voss, A conjugate-gradient approach to the parameter estimation problem of magnetic resonance advection imaging, *Inverse Problems in Science and Engineering* 28 (2020) 1154–1165.

[14] A.R. Hedar and M. Fukushima, Hybrid simulated annealing and direct search method for nonlinear unconstrained global optimization, *Optimization Methods and Software* 17 (2002) 891–912.

[15] M.R. Hestenes and E.L. Stiefel, Methods of conjugate gradients for solving linear systems, *Journal of Research of the National Bureau of Standards 49* (1952) 409–436.

[16] A.H. Ibrahim, P. Kumam, A.B. Abubakar, J. Abubakar and A. B. Muhammad, Least-square-based three-term conjugate gradient projection method for $l1$-norm problems with application to compressed sensing, *Mathematics* 8 (2020) 602.

[17] F.H. Jackson, On q-functions and a certain difference operator, *Earth and Environmental Science Transactions of the Royal Society of Edinburgh* 46 (1909) 253–281.

[18] D.O. Jackson, T. Fukuda, O. Dunn and E. Majors, On q-definite integrals, *Quarterly Journal of Pure and Applied Mathematics*, 1910.

[19] K.S. Kim and J.H. Yun, Image restoration using a fixed-point method for a TVL2 regularization problem, *Algorithms* 13 (2020) 1.

[20] N. Lin, Y. Chen and L. Lu, Mineral potential mapping using a conjugate gradient logistic regression model, *Natural Resources Research* 29 (2020) 173–188.

[21] J. Lin and C. Jiang, An improved conjugate gradient parametric detection based on space-time scan, *Signal Processing* 169 (2020) 107412.

[22] J. Liu, S. Du and Y. Chen, A sufficient descent nonlinear conjugate gradient method for solving M-tensor equations, *Journal of Computational and Applied Mathematics* 371 (2020) 112709.

[23] M. Fukushima, A conjugate gradient algorithm for sparse linear inequalities, *Journal of computational and applied mathematics* 30 (1990) 329–339.

[24] K.K. Lai, S.K. Mishra, G. Panda, S.K. Chakraborty, M.E. Samei, and B. Ram, A limited memory q-BFGS algorithm for unconstrained optimization problems, *Journal of Applied Mathematics and Computing* (2020) 1-20.

[25] S.K. Mishra and B. Ram, Steepest Descent Method, in: *Introduction to Unconstrained Optimization with R*, Springer: Singapore, 2019, pp. 131–173.

[26] S.K. Mishra and B. Ram, Newton's Method, in: *Introduction to Unconstrained Optimization with R, Springer, Singapore*, 2019, pp. 175–209.

[27] S.K. Mishra and B. Ram, Quasi-Newton Methods, in: *Introduction to Unconstrained Optimization with R*, Springer, Singapore, 2019, pp. 245–289.

[28] S.K. Mishra and B. Ram, *Conjugate Gradient Methods*, in: *Introduction to Unconstrained Optimization with R*, Springer, Singapore, 2019, pp. 211–244.

[29] I.H. Osman and J.P. Kelly, Meta-heuristics: an overview, in: *Meta-Heuristics*, Springer, Boston, MA, 1996, pp. 1–21.

[30] E. Polak, and G. Ribiére, Note sur la convergence de directions conjugées, *Rev. Francaise Informat Recherche Opertionelle* 3e Année 16 (1969) 35–43.

[31] B.T. Polyak, The conjugate gradient method in extreme problems, *USSR Computational Mathematics and Mathematical Physics* 9 (1969) 94–112.

[32] P.M. Rajković, S.D. Marinković and M.S. Stanković, On q-Newton-Kantorovich method for solving systems of equations, *Applied Mathematics and Computation* 168 (2005) 1432–1448.

[33] E. Yamakawa and M. Fukusima. A black-parallel conjugate gradient method for separable quadratic programming problems[1], *Journal of the Operations Research Society of Japan* 39 (1996) 407–427.

[34] A. Ahmadian, S. Rezapour, S. Salahshour and M.E. Samei, Solutions of sum - type singular fractional q integro - differential equation with m - point boundary value problem using quantum calculus., *Mathematical Methods in the Applied Sciences* 43 (2020), 8980–9004.

[35] G.N. Vanderplaats, Numerical Optimization Techniques for Engineering Design: with Applications, McGraw-Hill: Ryerson, 1984.

[36] G. Zoutendijk, *Nonlinear programming, computational methods*, in: *Integer and Nonlinear Programming*, J. Abadie (Eds.), North-Holland, Amsterdam, 1970, pp. 37–86.

[37] L. Zhang, W. Zhou and D. Li, Global convergence of a modified Fletcher-Reeves conjugate gradient method with Armijo-type line search, *Numerische Mathematik* 104 (2006) 561–572.

Manuscript received 10 June 2020
revised 12 December 2020
accepted for publication 12 January 2021

Kin Keung Lai
College of Economics, Shenzhen University
Shenzhen, 518060, China
E-mail address: mskklai@outlook.com

Shashi Kant Mishra
Department of Mathematics, Institute of Science
Banaras Hindu University, Varanasi, 221005, India
E-mail address: bhu.skmishra@gmail.com

Bhagwat Ram
DST-Centre for Interdisciplinary Mathematical Sciences
Institute of Science, Banaras Hindu University, Varanasi, 221005, India
E-mail address: bhagwatram14@gmail.com