Journal of Applied Mathematics and Computing (2020) 63:391–410
https://doi.org/10.1007/s12190-020-01322-x

ORIGINAL RESEARCH

# On $q$-Newton's method for unconstrained multiobjective optimization problems

**Shashi Kant Mishra**[1] · **Geetanjali Panda**[2] · **Md Abu Talhamainuddin Ansary**[3] · **Bhagwat Ram**[4]

Received: 26 November 2019 / Published online: 28 January 2020


## Abstract

In this paper, we present a method of so-called $q$-Newton's type descent direction for solving unconstrained multiobjective optimization problems. The algorithm presented in this paper is implemented by applying an independent parameter $q$ (quantum) in an Armijo-like rule to compute the step length which guarantees that the value of the objective function decreases at every iteration. The search processes gradually shift from global in the beginning to local as the algorithm converges due to $q$-gradient. The algorithm is experimented on 41 benchmark/test functions which are unimodal and multi-modal with 1, 2, 3, 4, 5, 10 and 50 different dimensions. The performance of the proposed method is confirmed by comparing with three existing schemes.

**Keywords** Optimization · Newton-type methods · $q$-calculus · Algorithms · Pareto optimality

**Mathematics Subject Classification** 78M50 · 49M15 · 05A30 · 68Wxx · 58E17

---

✉ Bhagwat Ram
bhagwatram14@gmail.com

Shashi Kant Mishra
bhu.skmishra@gmail.com

Geetanjali Panda
geetanjali@maths.iitkgp.ernet.in

Md Abu Talhamainuddin Ansary
md.abutalha2009@gmail.com

1 Department of Mathematics, Institute of Science, Banaras Hindu University, Varanasi, India

2 Department of Mathematics, Indian Institute of Technology, Kharagpur, India

3 Department of Economic Science, Indian Institute of Technology, Kanpur, India

4 DST-Centre for Interdisciplinary Mathematical Sciences, Institute of Science, Banaras Hindu University, Varanasi, India

## 1 Introduction

Multiobjective optimization has played an important role in solving real-world problems. Most engineering problems require the designer to optimize several conflicting objectives. The objectives are in conflict to each other if an improvement in one objective leads to the deterioration in another. In multiobjective optimization problems, several objective functions have to be minimized simultaneously. In this method, no single point can minimize all objective functions at a time. Therefore, the concept of optimality is replaced with Pareto optimality or efficiency [1]. A point is called Pareto optimal or efficient, if there does not exist a different point with the same or smaller objective function values, such that there is a decrease in at least one objective function value. Multiobjective unconstrained optimization problems have been applications in engineering design [2,3], design [4–6], location science [7], statistics [8], medicine [9–11], and cancer treatment planning [12], etc. There are many new studies on this field to solve the multiobjective unconstrained optimization problems [13–16]. A general solution approach for the multiobjective optimization problem is the scalarization technique which is widely used for computing the proper efficient solutions [17]. This method is free from priori chosen weighting factors or any other form of a prior ranking or ordering information for the different objective functions [18,19]. Several parameter dependent scalarization approaches for solving nonlinear multiobjective optimization problems are discussed in [20]. Scalarization techniques convert the original multiobjective optimization problem into a new single objective optimization problem in such a way that the optimal solution for the new problem is also optimal for the original one. From a practical point of view, the main advantage of this approach is that several fast and reliable methods developed for solving single objective optimization problems can be used to solve multiobjective optimization problems. In multiobjective optimization, one of the most widely used scalarization techniques is the weighting method, which consists of minimizing the weighted sum of different objectives [21]. In general, the weights, which are critical for the methods, are not known in advance for us. Thus, the computational implementations of this technique are not straightforward. Of course, the random choices of the weighting vector do not yield an optimal solution. The extension of the weighting method is for vector optimization [22] .

It is well known that the objective functions are minimized rapidly along the descent direction. The $q$-calculus was first developed by Jackson [23], and the results obtained in [24] rise to generalizations of series, functions and special numbers within the context of the $q$-calculus [25]. The $q$-calculus has been one of the research interests in the field of mathematics, physics, and signal processing for the last few decades [24,26]. The $q$-Newton-Kantorovich method [27] has been developed to solve the system of nonlinear equations such as:

$$|x_1^2 - 4| + e^{7x_2-36} - 2 = 0, \tag{1}$$

$$\log_{10}\left(\frac{12x_1^2}{x_2} - 6\right) + x_1^4 - 9 = 0. \tag{2}$$

With a starting point $x^0 = (2, 5)^T$, the solution $x^* = (\sqrt{3}, \frac{36}{7})^T$ is obtained. But, the classical Newton-Kantorovich method with the same starting point fails because the partial derivative of (1) with respect to the first variable does not exist. This is one of the motivation to use the $q$-derivative over the classical derivative. The $q$-derivative has been used in the steepest descent method to solve single objective unconstrained optimization problems [28]. It shows that the generated points are escaped from many local minima and reach to the global minima. Global optimization using $q$-gradient was further studied in [29], where the parameter $q$ is a dilation that is used to control the degree of the localness of the search. The $q$-derivative concept has also been used to develop $q$-least mean squares algorithm given in [30], which shows that the $q$-derivative takes larger steps to get the optimal solution for $q \in (0, 1)$ when compared to the conventional derivative. Recently, the $q$-derivative in the gradient of the given function is used to show the local convergent scheme, and then this idea extended to show the global convergence property for single objective unconstrained optimization [31]. The advantages of applying the $q$-derivative in multiobjective unconstrained optimization problems are given as follows:

1. When $q \neq 1$, the $q$-gradient vector can make any angle with the classical gradient vector, and the search direction can point in any direction. For example, for the case of the steepest descent method for single objective optimization problems, the descent direction can reduce the zigzag movement to obtain the optimal solution [28].
2. It minimizes the cost for solving multiobjective optimization problems because $q$-gradient takes the larger steps in the search direction as it evaluates the secant of the function rather than the tangent for the case of classical derivative [30].

To the best of our knowledge, the $q$-derivative has not been applied in Newton's method to solve multiobjective unconstrained optimization problems so far. In this paper, we apply the $q$-derivative to compute the $q$-Hessian which is used to find Newton's search direction, and generalize the algorithm given in [32] and prove the convergence theorem.

The outline of this paper is organized in the following manner: in Sect. 2, some prerequisites related to the multiobjective optimization problems are discussed. In Sect. 3, we present the first-order optimality condition for multiobjective unconstrained optimization using $q$-derivative, and present Newton's search descent direction. In Sect. 4, we give the algorithm with convergence theorem, and numerical examples are given in Sect. 5. The last section is conclusion.

## 2 Preliminaries

We address the following multiobjective unconstrained optimization problem (MUOP):

$$\min \quad f(x), \quad x \in \mathbb{R}^n, \tag{3}$$

where $f : \mathbb{R}^n \to \mathbb{R}^m$, $f(x) = (f_1(x), f_2(x), \dots, f_m(x))$, and $f_j : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable for $j = 1, \dots, m$. For $x, y \in \mathbb{R}^n$ we denote vector inequalities as

$$
\begin{aligned}
x = y &\iff x_i = y_i \text{ for all } i = 1, \ldots, n, \\
x \geqq y &\iff x_i \geq y_i \text{ for all } i = 1, \ldots, n, \\
x \geq y &\iff x_i \geq y_i \text{ and } x \neq y, \\
x > y &\iff x_i > y_i \text{ for all } i = 1, \ldots, n.
\end{aligned}
$$

The $q$-derivative ($q \neq 1$) of $f_j$ for $j = 1, \ldots, m$ is defined as

$$
D_{q,x} f_j(x) = \begin{cases} \frac{f_j(x) - f_j(qx)}{x(1-q)}, & x \neq 0, \\ f_j'(x), & x = 0. \end{cases} \tag{4}
$$

In the limit as $q \to 1$ or $x \to 0$, the $q$-derivative reduces to the classical derivative. Suppose the partial derivatives of $f_j : \mathbb{R}^n \to \mathbb{R}$, for $j = 1, \ldots, m$ exist. For $x \in \mathbb{R}^n$, consider an operator $\epsilon_{q,i}$ on $f_j$ as

$$
(\epsilon_{q,i} f_j)(x) = f_j(x_1, x_2, \ldots, x_{i-1}, qx_i, x_{i+1}, \ldots, x_n). \tag{5}
$$

The $q$-partial derivative ($q \neq 1$) of $f_j$ for $j = 1, \ldots, m$ at $x$ with respect to $x_i$ for $i = 1, \ldots, n$ is

$$
D_{q,x_i} f_j(x) = \begin{cases} \frac{f_j(x) - (\epsilon_{q,i} f_j)(x)}{(1-q)(x_i)}, & x_i \neq 0, \\ \frac{\partial f_j}{\partial x_i}, & x_i = 0. \end{cases} \tag{6}
$$

We denote

$$
g_j(x) = \nabla f_j(x) = (g_1^j(x), g_2^j(x), \ldots, g_n^j(x))^T,
$$

where $g_i = \frac{\partial f_j}{\partial x_i}$ for $i = 1, \ldots, n$, $j = 1, \ldots, m$. The Jacobian of the function $f_j$ for $j = 1, \ldots, m$ is the $q$-partial derivative of $\nabla f_j(x)$, which is given as:

$$
D_q \nabla f_j(x) = \begin{bmatrix} D_{q,x_1} g_1^j(x) & D_{q,x_2} g_1^j(x) & \ldots, & D_{q,x_n} g_1^j(x) \\ D_{q,x_1} g_2^j(x) & D_{q,x_2} g_2^j(x) & \ldots & D_{q,x_n} g_2^j(x) \\ \cdots & \cdots & \cdots & \cdots \\ D_{q,x_1} g_n^j(x) & D_{q,x_2} g_n^j(x) & \ldots & D_{q,x_n} g_n^j(x) \end{bmatrix}_{n \times n}. \tag{7}
$$

In short, we write $D_q \nabla f_j(x) = [D_{q,x_i} g_i(x)]_{n \times n}, \ \forall \ i = 1, \ldots n$, and $j = 1, \ldots, m$. The matrix $D_q \nabla f_j(x)$ is not necessarily a symmetric matrix. For example, let $f : \mathbb{R}^2 \to \mathbb{R}$ be a function defined by

$$
f(x) = 3x_1^2 - 5x_1 x_2^3. \tag{8}
$$

Then,

$$\nabla f(x) = [6x_1 - 5x_2^3, -15x_1x_2^2]^T,$$

and

$$D_q\nabla f(x) = \begin{bmatrix} 6 & -5(1+q+q^2)x_2^2 \\ -15x_2^2 & -15(1+q)x_1x_2 \end{bmatrix},$$

which is not symmetric. A point $x^* \in X$ is a Pareto optimum, if there is no $y \in X$ for which $f_j(y) \leq f(x^*)$, $j = 1, \ldots, m$, and $f(y) \neq f(x^*)$. The point $x^* \in X$ is a weak Pareto optimum, if there is no $y \in X$ for which $f_j(y) < f(x^*)$, $j = 1, \ldots, m$. Let $\mathbb{R}_{++}$ be the set of strictly positive real numbers. Assume that $X \subseteq \mathbb{R}^n$ is an open set and $f_j : X \to \mathbb{R}$, $j = 1, \ldots, m$ is given function. The directional derivative of $f_j$, where $j = 1, \ldots, m$ at $x \in \mathbb{R}^n$ in the direction $d \in \mathbb{R}^n$ is defined as

$$f_j'(x; d) = \lim_{\alpha \to 0} \frac{f_j(x + \alpha d) - f_j(x)}{\alpha}, \; \forall \, j = 1, \ldots, m. \tag{9}$$

For $x \in \mathbb{R}^n$, $\|x\|$ denotes the Euclidean norm in $\mathbb{R}^n$. Norm of a matrix $A \in \mathbb{R}^{n\times n}$ is $\|A\| = \max \frac{\|Ax\|}{\|x\|}$, $x \neq 0$. We say $d \in \mathbb{R}^n$ as a descent direction for $f_j$ at $x$, if for all $j = 1, \ldots, m$, $d^T\nabla f_j(x) < 0$. Thus, $d$ is a descent direction of $f$ at $x$, if there exists $\alpha_0 > 0$ such that $f_j(x + \alpha d) < f_j(x)$ for all $\alpha \in (0, \alpha_0]$. A point $x^* \in \mathbb{R}^n$ is said to be an efficient point of (MUOP), if there does not exist $y \in \mathbb{R}^n$ such that $f_j(y) \leq f_j(x^*)$, $j = 1, \ldots, m$. This means that point $x^*$ is weak efficient point of $f_j$, if there does not exist $d \in \mathbb{R}^n$ such that $g_j(x^*)^T d < 0$ for all $j = 1, \ldots, m$. The next proposition due to [32] establishes the relationship between the properties of being a critical and an optimal point.

**Proposition 1** *Let $f_j$, $j = 1, \ldots, m$, be a continuously differentiable on $X \subset \mathbb{R}^n$. Then,*

1. *If $x^*$ is locally weak Pareto optimal, then $x^*$ is a critical point for $f_j$.*
2. *If $f_j \in \mathbb{R}^m$ is convex, and $x^*$ is a critical for $f_j$, then $x^*$ is a weak Pareto optimal.*
3. *If $f_j \in C^2(\mathbb{R}^n, \mathbb{R}^m)$, and Hessian matrices are positive definite for all $x \in \mathbb{R}^n$, and if $x^*$ is critical point for all $f_j$, then $x^*$ for all $j = 1, \ldots, m$, is a Pareto optimal.*

The idea of our proposed algorithm is very straightforward: choose an initial guess $x^0$ and check if part 1 of Proposition 1 holds. If not, compute a Newton search direction and make a suitable step length from $x^0$ along Newton search direction, which results a new point and the algorithm is repeated in this way.

## 3 On $q$-Newton type descent direction

We now proceed to present the $q$-Newton descent direction for multiobjective unconstrained optimization problem. For any point $x \in \mathbb{R}^n$, we denote $d_{Nq}(x)$, the Newton direction as the optimal solution of the following problem:

$$\begin{cases} \min \quad \max_{j=1,\ldots,m} g_j(x)^T d_{Nq}(x) + \frac{1}{2} d_{Nq}(x)^T D_q \nabla f_j(x) d_{Nq}(x) \\ \text{subject to} \ \ d_{Nq}(x) \in \mathbb{R}^n. \end{cases} \tag{10}$$

We consider the symmetric counter part $\bar{D}_q$ of $D_q$ as

$$\bar{D}_q = \frac{1}{2}(D_q + D_q^T). \tag{11}$$

In addition to this, $D_q \nabla f_j$, $j = 1, \ldots, m$ may not be positive definite for some $q$. Here, we assume the symmetric counter part $\bar{D}_q \nabla f_j$, $j = 1, \ldots, m$, and the positive definiteness of $\bar{D}_q \nabla f$ is in the vicinity of $x^*$. The optimal value of problem (10) is given as:

$$\theta(x) : \inf_{d_{Nq} \in \mathbb{R}^n} \max_{j=1,\ldots,m} g_j(x)^T d_{Nq} + \frac{1}{2} d_{Nq}^T \bar{D}_q \nabla f_j(x) d_{Nq}. \tag{12}$$

For multiobjective optimization, Newton search direction [32] is obtained by minimizing the maximum of quadratic term, which is given as:

$$d_{Nq}(x) : \arg\min_{d_{Nq} \in \mathbb{R}^n} \max_{j=1,\ldots,m} \ g_j(x)^T d_{Nq} + \frac{1}{2} d_{Nq}^T \bar{D}_q \nabla f_j(x) d_{Nq}. \tag{13}$$

The problem (10) is a non-smooth problem, but it also involves quadratic approximation of each objective function.

The above problem will be a quadratic convex programming problem, if every objective function is strongly convex. Therefore, such problem always has a unique minimizer, which is presented as:

$$P(x) : \begin{cases} \min \qquad \Gamma(x) \\ \text{subject to} \ \ g_j(x)^T d_{Nq}(x) + \frac{1}{2} d_{Nq}^T \bar{D}_q \nabla f_j(x) d_{Nq}(x) \leq \Gamma(x), \ \ 1 \leq j \leq m, \\ \qquad\qquad (\Gamma(x), d_{Nq}) \in \mathbb{R} \times \mathbb{R}^n. \end{cases}$$

Thus,

$$\Gamma(x) : \arg\min_{d_{Nq} \in \mathbb{R}^n} \max_{j=1,\ldots,m} \ g_j(x)^T d_{Nq} + \frac{1}{2} d_{Nq}^T \bar{D}_q \nabla f_j(x) d_{Nq}. \tag{14}$$

Also, note that for $m = 1$, the Newton direction $d_{Nq}(x)$ becomes the classical Newton direction for scalar optimization problems.

For $x \in \mathbb{R}^n$, necessary condition for Pareto optimality is given in [1], and defined for steepest descent like methods for multiobjective case in [33,34], which is modified as:

$$\Re(\bar{D}_q \nabla f_j(x)) \cap (-\mathbb{R}^m_{++}) = \phi, \ \ \forall \ j = 1, \ldots, m. \tag{15}$$

Note that $P(x)$ has unique solution, which can be obtained using Karush-Kuhn-Tucker (KKT) optimality conditions. The Lagrange function of problem $P(x)$ is:

$$L(\Gamma, d_{Nq}; \lambda) = \Gamma(x) + \sum_{j=1}^{m} \lambda_j \left( g_j(x)^T d_{Nq} + \frac{1}{2} d_{Nq}^T \bar{D}_q \nabla f_j(x) d_{Nq} - \Gamma(x) \right), \tag{16}$$

where $\lambda_j \geq 0$ are Lagrange multipliers. The corresponding KKT optimality conditions for $P(x)$ are given as:

$$\sum_{j=1}^{m} \lambda_j \big( g_j(x)^T + \bar{D}_q \nabla f_j(x) d_{Nq}(x) \big) = 0, \tag{17}$$

$$\sum_{j=1}^{m} \lambda_j = 1, \tag{18}$$

$$\lambda_j \geq 0, \quad g_j(x)^T d_{Nq}(x) + \frac{1}{2} d_{Nq}(x)^T \bar{D}_q \nabla f_j(x) d_{Nq}(x) \leq \Gamma(x), \quad \forall\ j = 1, \ldots, m, \tag{19}$$

$$\lambda_j \left( g_j(x)^T d_{Nq}(x) + \frac{1}{2} d_{Nq}^T \bar{D}_q \nabla f_j(x) d_{Nq}(x) - \Gamma(x) \right) = 0,\ \forall\ j = 1, \ldots, m. \tag{20}$$

Suppose $d_{Nq}(x)$ satisfies (17)–(20) with Lagrange multipliers $\lambda_j$, where $j = 1, \ldots, m$. The optimal value of $P(x)$ is $\Gamma(x)$. In particular, from (17), we obtain following:

$$d_{Nq}(x) = - \left[ \sum_{j=1}^{m} (\bar{D}_q \nabla f_j(x) \right]^{-1} \sum_{j=1}^{m} \lambda_j g_j(x)^T. \tag{21}$$

**Theorem 1** *For any noncritical point $x \in \mathbb{R}^n$, the Newton direction $d_{Nq}(x)$, as defined in (21) is a descent direction at $x$.*

***Proof*** Note that, from (14), for $d_{Nq}(x) = 0$, we have $\Gamma(x) \leq 0$. Suppose $x$ is not a critical point of $f_j, \forall\ j = 1, \ldots, m$, then we must have

$$\Re(\bar{D}_q \nabla f_j(x)) \cap (-\mathbb{R}^m_{++}) \neq \phi, \ \ \forall\ j = 1, \ldots, m.$$

Thus, there exists $d(x) \in \mathbb{R}^n$ such that $g_j(x)^T d_{Nq}(x) < 0, \forall\ j = 1, \ldots, m$. Replacing $d(x)$ by $\gamma d_{Nq}(x)$ for any $\gamma \in (0, 1)$, we get

$$\Gamma(x) \leq \max_{j=1,\ldots m} g_j(x)^T \gamma d(x) + \frac{1}{2}\gamma d(x)^T \bar{D}_q \nabla f_j(x) \gamma d(x)$$
$$= \gamma \max_{j=1,\ldots m} g_j(x)^T d(x) + \frac{1}{2} d(x)^T \bar{D}_q \nabla f_j(x) d(x).$$

Therefore, for $\gamma > 0$, $\max_{j=1,\ldots,m} g_j(x)^T d_{Nq}(x) + \frac{1}{2} d_{Nq}(x)^T \bar{D}_q \nabla f_j(x) d_{Nq}(x)$ is negative, that is, $\Gamma(x) < 0$. Since $\bar{D}_q \nabla f_j(x)$ is positive definite matrix, and $d_{Nq}(x) \neq 0$, then

$$g_j(x)^T d_{Nq}(x) < g_j(x)^T d_{Nq}(x)^T \bar{D}_q \nabla f_j(x) d_{Nq}(x) \leq \Gamma(x) < 0. \tag{22}$$

Thus, $g_j(x)^T d_{Nq}(x) < 0, \forall\ j = 1, \ldots, m$. Thus, $d_{Nq}(x)$ is a descent direction. This completes the proof. □

***Remark 1*** We say that $x^*$ is a critical point of $f_j$ where $j = 1, \ldots, m$ if and only if $\Gamma(x^*) = 0$, and $d_{Nq}(x^*) = 0$.

**Theorem 2** *Let function $d_{Nq} : X \to \mathbb{R}$ given by (21) be a bounded on compact sets and $\Gamma : X \to \mathbb{R}$ given by (14), then $|\Gamma(x) - \Gamma(y)| < \epsilon, \forall\ \ x, y \in X$.*

***Proof*** Let $Y \subset X$ be a compact set for any $x \in X$, and we have $\Gamma(x) \leq 0$ due to part 1 in Lemma 3.2 of [32], then

$$g_j(x)^T d_{Nq}(x) + \frac{1}{2} d_{Nq}(x)^T \bar{D}_q \nabla f_j(x) d_{Nq}(x) \leq 0.$$

We obtain

$$-\frac{1}{2} d_{Nq}(x)^T \bar{D}_q \nabla f_j(x) d_{Nq}(x) \geq g_j(x)^T d_{Nq}(x). \tag{23}$$

Note that $f_j$ is twice continuously differentiable, and its all $q$-Hessians are positive definite due to (11), so there exists $\kappa$ and $\lambda > 0$ such that

$$\kappa = \max_{x \in Y, j=1,\ldots,m} \|g_j(x)\|, \tag{24}$$

and

$$\lambda = \min_{y \in Y, \|\theta\|=1} \theta^T \bar{D}_q \nabla f_j(y) \theta, \tag{25}$$

where $j = 1, \ldots, m$. Combining (23)–(25) and using Cauchy–Schwartz inequality, for $x \in Y$, and $j = 1, \ldots, m$, we get

$$\lambda d_{Nq}(x)^T d_{Nq}(x) \leq \|g_j(x)\| \|d_{Nq}(x)\| \leq \kappa \|d_{Nq}(x)\|. \tag{26}$$

Therefore,

$$\|d_{Nq}(x)\| = \frac{1}{\lambda}\kappa, \tag{27}$$

for all $y \in Y$. As any point in $Y$ is in the interior of a compact subset of $Y$, then it suffices to show that continuity of $\Gamma(x)$ on an arbitrary compact set $Y \subset X$. For $y \in Y$, and $\phi_{y,j} : Y \to \mathbb{R}$, where $j = 1, \dots, m$ such that

$$\zeta \to g_j(\zeta)^T d_{Nq}(y) + \frac{1}{2} d_{Nq}(x)^T \bar{D}_q \nabla f_j(\zeta) d_{Nq}(y). \tag{28}$$

The family $\{\psi_{y,j}\} \in Y,\ j = 1, \dots, m$ is equi-continuous.

$$\Gamma(\zeta) \leq \max_{j=1,\dots m} g_j(\zeta)^T \gamma d_{Nq}(y) + \frac{1}{2}\gamma d_{Nq}^T(y) \bar{D}_q \nabla f_j(\zeta) \gamma d_{Nq}(y),$$

that is,

$$\Gamma_y(\zeta) \leq \phi_y + |\Gamma_y(\zeta) - \Gamma_y(y)| + \epsilon.$$

Thus, $|\Gamma(\zeta) - \Gamma(y)| < \epsilon$. Interchanging the roles of $\zeta$ and $y$, we conclude that $\Gamma$ is continuous on $X$.
This completes the proof. □

## 4 On *q*-Newton unconstrained multiobjective algorithm and convergence

On the basis of theory described in previous section, we present the algorithm of $q$-Newton unconstrained multiobjective algorithm for solving (MOUP) using q-derivative. We examine $\Gamma(x)$ to obtain the Newton direction $d_{Nq}(x)$ at each non-critical point. The step length is determined by means of inexact Armijo condition with back-tracking line search method. The algorithm for finding a critical/Pareto front is given below.

We now present the convergence theorem of Algorithm 1. Observe that if Algorithm 1 terminates after a finite number of iterations, then it terminates at a Pareto critical point. The following theorem is the modification of [35].

**Theorem 3** *Let $f_j$ be continuously differentiable on a compact set $X \subset \mathbb{R}^n$ for $j = 1, \dots, m$ and $\{x^k\}$ be the sequence by $x^{k+1} = x^k + \alpha_k d_{Nq}(x^k)$ given in Algorithm 1, and $\alpha_k$ satisfies*

$$f_j\left(x^{k+1}\right) - f_j\left(x^k\right) \leq c\alpha_k \sum_{j=1}^{m} \lambda_j^k \left(g_j(x^k)\right)^T d_{Nq}\left(x^k\right), \tag{29}$$

**Algorithm 1:** $q$-Newton Unconstrained Multiobjective Algorithm

Choose $x^0 \in X \subseteq \mathbb{R}^n$, error of tolerance $\epsilon > 0$, fix $q \in (0, 1)$, small positive number $\delta$ such that $0 < \delta < 1$, $\theta_j^k$ is the angle between $g_j(x^k)$ and $d_{Nq}(x^k)$ ;
**for** *k=0,1,2,...* **do**
  Compute $\Gamma(x^k)$, and $d_{Nq}(x^k)$;
  **for** $j = 1, \ldots, m$ **do**
    **if** $\cos^2(\theta_j^k) \geq \delta$ **then**
      then choose appropriate step length $\alpha_k$ such that $x^k + \alpha_k d_{Nq}(x^k) \in X$ and satisfies (29) and (31).
    **end**
    **if** $\cos^2(\theta_j^k) < \delta$ **then**
      choose appropriate step length $\alpha_k$ such that $x^k + \alpha_k d_{Nq}(x^k) \in X$ and satisfies (29).
    **end**
  **end**
  Set $x^{k+1} = x^k + \alpha_k d_{Nq}(x^k)$;
  **if** $\Gamma(x^{k+1}) < \epsilon$ *or* $\|g_j^{k+1}\| < \epsilon$ **then**
    stop;
  **end**
**end**

*for all $j = 1, \ldots, m$. Suppose that $L_0 = \{x \in X : f(x) < f(x^0)\}$ is bounded and convex, where $x^0 \in X$ is an initial guess point. The function $f_j(x)$ is bounded below for at least one $j \in \{1, \ldots, m\}$. Then, the accumulation point of $\{x^k\}$ is a critical point of $x^*$ of (MOUP).*

***Proof*** We have

$$f_j\left(x^{k+1}\right) - f_j\left(x^k\right) \leq c\alpha_k \sum_{i=1}^{m} \lambda_j^k \left(g_j(x^k)\right)^T d_{Nq}\left(x^k\right).$$

Since $\sum_{j=1}^m \lambda_j^k = 1$, and $\lambda_j^k \geq 0$, then

$$f_j\left(x^{k+1}\right) - f_j\left(x^k\right) \leq c\alpha_k \max_{j=1,\ldots m} \left((g_j(x^k))^T d_{Nq}(x^k)\right).$$

Since $\bar{D}_q \nabla f_j(x)$ is positive definite, then

$$f_j\left(x^{k+1}\right) - f_j\left(x^k\right) < c\alpha_k \max_{j=1,\ldots m} \left((g_j(x^k))^T d_{Nq}(x^k) + \frac{1}{2} d_{Nq}\left(x^k\right)^T \bar{D}_q \nabla f_j(x) d_{Nq}\left(x^k\right)\right) = c\alpha_k \Gamma\left(x^k\right).$$

We obtain

$$f_j\left(x^{k+1}\right) < f_j\left(x^0\right) + c\sum_{i=0}^{k} \alpha_i \Gamma\left(x^i\right) \quad \text{for all } j = 1, \ldots, m.$$

Fix one $j_1$ from $j = 1, \ldots, m$ for which $f_j(x)$ is bounded below such that $f_{j1}(x) > -\infty$ for all $x \in X$. Also, $\{f_{j1}(x^k)\}$ is monotonically decreasing sequence which is bounded below where $f_{j1}(x^*) > -\infty$. Thus,

$$f_{j1}\left(x^0\right) - f_{j1}\left(x^{k+1}\right) > -c\sum_{i=0}^{k} \alpha_i \Gamma\left(x^i\right).$$

Taking $k \to \infty$ in the above inequality to get following:

$$c\sum_{i=0}^{\infty} \alpha_i \left(-\Gamma(x^i)\right) < f_{j1}\left(x^0\right) - f_{j1}\left(x^*\right) < \infty \tag{30}$$

We already know that $\Gamma(x^i) \le 0$ for all $i$, and $c\sum_{i=0}^{\infty} \alpha_i(-\Gamma(x^i))$ is finite. Thus, we obtain $c\alpha_k(-f(x^k)) \to 0$ as $k \to \infty$. Since the step length is bounded above so $\alpha_k \to \infty$ for some $k$ implies $L_0$ unbounded which is contradiction to the assumption. If $\alpha_k \ge \beta$ for all $k$ and for some $\beta > 0$, then we get $-f(x^k) \to 0$ as $k \to \infty$. Note that $L_0$ is bounded sequence, and has at least one accumulation point. Let $\{P_1^*, P_2^*, \ldots, P_r^*\}$ be the set of accumulation points $\{x^k\}$. Since $P_s^*$ is an accumulation point for every $s \in \{1, 2, \ldots, r\}$, and $\Gamma$ is a continuous function, then $\Gamma(P_s^*)$ is a critical point of $f$ for every $s \in \{1, 2, \ldots, r\}$.
This completes the proof. □

**Theorem 4** *Let $f_j$ be a continuously differentiable on a compact set $X \subset \mathbb{R}^n$ for every $j = 1, \ldots, m$, and $\{x^k\}$ be the sequence by $x^{k+1} = x^k + \alpha_k d^k_{Nq}(x^k)$, and given that*

1. $$c_2\sum_{j=1}^{m} \lambda_j^k \left(g_j(x^k)\right)^T d_{Nq}\left(x^k\right) \le \sum_{j=1}^{m} \lambda_j^k g_j\left(x^{k+1}\right) d_{Nq}\left(x^k\right), \tag{31}$$
2. *$g_j$ are Lipschitz for all $j = 1, \ldots, m$, and*
3. *$\cos^2 \theta_j^k \ge \delta$ for some $\delta > 0$, for all $j = 1, \ldots, m$, where $\theta_j^k$ is the angle between $d_{Nq}(x^k)$ and $g_j(x^k)$.*

*Then, every accumulation point of $\{x^k\}$ generated by Algorithm 1 is a weak efficient solution of (MOUP).*

***Proof*** From Theorem 3, we observe that every accumulation point of $\{x^k\}$ is a critical point of $f_j$, where $j = 1, \ldots, m$. Let $x^*$ be an accumulation point of $\{x^k\}$. Fix one $j_0$ from $j = 1, \ldots, m$ for which $g_{j_0}(x^*) = 0$, then $x^*$ will be a weak efficient solution. From part 2 of Theorem 4, $g_j$ are Lipschitz continuous for all $j = 1, \ldots m$. Thus, there exists $L_j > 0$ such that $\|g_j(x) - g_j(y)\| \le L_j\|x - y\|$ for $j = 1, \ldots, m$. Form Cauchy–Schwartz inequality, we have

$$\begin{aligned}\left(g_j(x^{k+1}) - g_j(x^k)\right)^T d_{Nq}\left(x^k\right) &\le \|g_j\left(x^{k+1}\right) - g_j\left(x^k\right)\|\|d_{Nq}\left(x^k\right)\| \\ &\le L_j\|x^{k+1} - x^k\|\|d_{Nq}(x^k)\| \\ &\le L_j\alpha_k\|d_{Nq}(x^k)\|^2,\end{aligned}$$

Since $L = \max L_j$, where $j = 1, 2 \ldots, m$, then

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

Thus,

$$\begin{aligned} L\alpha_k \left\| d_{Nq}\left(x^k\right)\right\|^2 &\geq \max_{j=1,\ldots m} \left(g_j(x^{k+1}) - g_j(x^k)\right)^T d_{Nq}\left(x^k\right) \\ &\geq \sum_{j=1}^{m} \lambda_j^k \left(g_j(x^{k+1}) - g_j(x^k)\right)^T d_{Nq}\left(x^k\right). \end{aligned}$$

From part 1 of Theorem 4, we get

$$\begin{aligned} L\alpha_k \left\| d_{Nq}(x^k)\right\|^2 &\geq (c_2 - 1) \sum_{j=1}^{m} \lambda_j^k g_j\left(x^k\right)^T d_{Nq}\left(x^k\right) \\ &\geq (c_2 - 1) \max_{j=1,\ldots,m} g_j\left(x^k\right)^T d_{Nq}\left(x^k\right). \end{aligned}$$

This implies

$$\alpha_k \geq \frac{c_2 - 1}{L\|d_{Nq}(x^k)\|^2} \max_{j=1,\ldots,m} g_j\left(x^k\right)^T d_{Nq}\left(x^k\right).$$

Since $\max_{j=1,\ldots,m} g_j(x^k)^T d_{Nq}(x^k) < 0$, then

$$\alpha_k\left(\max_{j=1,\ldots,m} g_j(x^k)^T d_{Nq}(x^k)\right) \leq \frac{c_2 - 1}{L\|d_{Nq}(x^k)\|^2}\left(\max_{j=1,\ldots,m} g_j(x^k)^T d_{Nq}(x^k)\right)^2,$$

that is,

$$-c_1\alpha_k\left(\max_{j=1,\ldots,m} g_j(x^k)^T d_{Nq}(x^k)\right) \geq \frac{c_1(c_2 - 1)}{L\|d_{Nq}(x^k)\|^2} \min_{j=1,\ldots,m} \left(g_j(x^k)^T d_{Nq}(x^k)\right)^2.$$

Since $(g_j(x^k)^T d_{Nq}(x^k))^2 = (g_j(x^k)^T)^2 (d_{Nq}(x^k))^2 (\cos^2 \theta_j^k)$, then

$$-c_1\alpha_k\left(\max_{j=1,\ldots,m} g_j(x^k)^T d_{Nq}(x^k)\right) \geq \frac{c_1(1 - c_2)}{L} \min_{j=1,\ldots,m} [\|g_j(x^k)\|^2 \cos^2(\theta_j)^T],$$

where $\theta_j^k$ is the angle between $g_j(x^k)$ and $d_{Nq}(x^k)$. We follow the same process done in Theorem 3.

$$\infty > f_{j1}\left(x^0\right) - f_{j1}\left(x^{k+1}\right) \geq -c_1 \sum_{i=0}^{k} \alpha_i \max_{y\in Y, j=1,\ldots,m} \left(g_j(x^i)\right)^T d_{Nq}\left(x^i\right)$$

$$= \sum_{i=0}^{k} \alpha_i \left(-c_1 \max_{y\in Y, j=1,\ldots,m} (g_j(x^i))^T d_{Nq}(x^i)\right).$$

Taking $k \to \infty$, we get

$$\infty > f_{j1}\left(x^0\right) - f_{j1}\left(x^*\right) \geq \sum_{i=0}^{\infty} \alpha_i \left(-c_1 \max_{j=1,\ldots,m} (g_j(x^i))^T d_{Nq}(x^i)\right).$$

Since $-c_1 \max_{j=1,\ldots,m} g_j(x^i)^T d_{Nq}(x^i) > 0$, then

$$\alpha_k(-c_1 \max_{j=1,\ldots,m} (g_j(x^k))^T d_{Nq}(x^k)) \to 0 \text{ as } k \to \infty.$$

We also have

$$\frac{c_1(1-c_2)}{L} \min_{j=1,\ldots,m} [\|g_j(x^k)\|^2 \cos^2\theta_j^k] \to 0 \quad \text{as} \quad k \to \infty.$$

Since $\cos^2\theta_j^k > \delta$ for $j = 1, \ldots, m$, then $\min_{j=1,\ldots,m} \|g_j(x^k)\|^2 \to 0$ as $k \to \infty$. Fix any $j_0$ from $j = 1, \ldots, m$ such that $\|g_{j0}(x^k)\|^2 \to 0$ as $k \to \infty$. Since $\|g_{j0}(x^k)\|$ is a continuous function, and $\|g_{j0}(x^k)\| \to 0$ as $k \to \infty$, then $g_{j0}(x^*) = 0$ for every accumulation points $x^*$ of $\{x^k\}$. Thus, $x^*$ is a local weak efficient solution.
This completes the proof. □

***Remark 2*** Algorithm 1 is also applicable to non-convex functions. It is important to note that weak efficient solution of any multiobjective unconstrained optimization problem is not unique. Thus, if Algorithm 1 is executed with any initial point, then the users may obtain any one out of these weak efficient points while shifting the descent direction from global to local rapidly due to $q$-gradient. All three assumptions of Theorem 4 should be satisfied for every accumulation point of the sequence $\{x^k\}$ to be a weak efficient point of (MOUP), otherwise accumulation point becomes critical point if assumptions of Theorem 3 is satisfied.

## 5 Numerical examples

In this section, Algorithm 1 is verified and compared with existing methods using some numerical problems from different sources. MATLAB (2019a) code is developed for Algorithm 1. To avoid unbounded solutions, the following subproblem is solved:

$$\bar{P}(x^k):\begin{cases}\min & \Gamma(x^k)\\ \text{subject to} & g_j(x^k)^T d_{Nq}(x)+\frac{1}{2}d_{Nq}^T\bar{D}_q\nabla f_j(x)d_{Nq}(x^k)\le\Gamma(x^k),\\ & lb\le x^k+d_{Nq}\le ub,\\ & (\Gamma(x^k),d_{Nq})\in\mathbb{R}\times\mathbb{R}^n,\end{cases}$$

where $j=1,\dots,m$, $lb$ and $ub$ are lower and upper bounds of $x$. Solution of $\bar{P}(x^k)$ is not a descent direction, if $\bar{D}_q\nabla f_j(x^k)$ is not positive definite for all $j$. In such cases, an approximation $\tilde{D}_q\nabla f_j(x^k)=\bar{D}_q\nabla f_j(x^k)+E(x^k)$ is used, where $E(x^k)$ is a diagonal matrix obtained using modified Cholesky factorization algorithm developed in [36]. The subproblem $\bar{P}(x^k)$ is solved using MATLAB function '*fmincon*' with '*Algorithm interior point*', '*Specified Objective Gradient*','*Specified Constraint Gradient*'. Also, $|\Gamma(x^k)|<10^{-5}$ or maximum 200 iterations is considered as stopping criteria.

It is important to note that weak efficient solution of a multiobjective optimization problem is not unique. Thus, if the users start at any initial point and execute the algorithm, then user may reach at one of weak efficient points. The weighting method is one of the most attractive procedures for solving multiobjective optimization problems. This is due to the fact that it reduces the original problem to a family of scalar minimization problems. We first verify the steps of Algorithm 1 for obtaining a critical point with the following example:

***Example 1*** Consider the multiobjective optimization problem:
$\min_{x\in\mathbb{R}^2}(f_1(x),f_2(x))$, where

$$f_1(x)=\begin{cases}(x_1-1)^3\sin\frac{1}{x_1-1}+(x_1-1)^2+x_1(x_2-1)^4, & \text{if } x_1\ne 1,\\ (x_2-1)^4, \ \text{if } \ x_1=1.\end{cases}$$

$f_2(x)=x_1^2+x_2^2.$

Note that $\frac{\partial^2 f_1}{\partial x_i^2}$; $(i=1,2)$ does not exist at a point $(1,1)^T$, which indicates that $f_1$ is not twice differentiable. Thus, second order sufficient condition can not be applied to justify the existence of the minimizer as in the case of higher order numerical optimization methods. Further, the Newton's algorithm can not be applied. But, the $q$-derivative can be applied as described below. For $q\ne 1$,

$$\bar{D}_q\nabla f_1(1,1)=\begin{bmatrix}3(q-1)\sin\frac{1}{q-1} & \frac{(q-1)^3}{2}\\ \frac{(q-1)^3}{2} & 4(q-1)^2\end{bmatrix}.$$

Note that $\bar{D}_q\nabla f_1(1,1)$ is positive definite when the principal minors are positive, that is,

$$3(q-1)\sin\frac{1}{q-1}>0,$$

and

$$det(\bar{D}_q \nabla f_1(1,1)) = 12 \sin \frac{1}{q-1} - \frac{(q-1)^3}{4} > 0.$$

In particular one may observe that, for

$$q \in (0,1) \cap \left(1 + \frac{1}{2k\pi}, 1 + \frac{1}{(2k+1)\pi}\right),$$

where $k \in \mathbb{Z}^-$, the above two inequalities hold. Therefore, for this selection of $q$, $\bar{D}_q \nabla f_1(1,1)$ is positive definite. We have solved the problem using Algorithm 1 with approximation $x^0 = (1.6, 1.5)^T$, initial parameters $q = 0.93$, $c_1 = 10^{-4}$, $c_2 = 0.9$, $\delta = 10^{-3}$ and error of tolerance $\epsilon = 10^{-5}$. We obtain $f(x^0) = (f_1(x^0), f_2(x^0)) = (0.6750, 4.81)^T$, $g_1(x^0) = (2.1491, 0.5814)^T$, $g_2(x^0) = (3.0880, 2.8950)^T$, $\bar{D}_q \nabla f_1(x^0) = \begin{bmatrix} 4.4794 & 0.3634 \\ 0.3634 & 2.9585 \end{bmatrix}$ and $\bar{D}_q \nabla f_2(x^0) = \begin{bmatrix} 1.93 & 0 \\ 0 & 1.93 \end{bmatrix}$. Both $\bar{D}_q \nabla f_1(x^0)$ and $\bar{D}_q \nabla f_2(x^0)$ are positive definite and hence solution of $P(x^0)$ is a descent direction of $f$. Solution of $P(x^0)$ is obtained as $\Gamma(x^0) = -0.5438$ and $d_{Nq}(x^0) = (-0.4685, -0.1390)^T$. Since $\cos^2(\theta_1^0) = 0.9994 > \delta$ and $\cos^2(\theta_2^0) = 0.7991 > \delta$ with $\alpha_0 = 1$ satisfying (29) and (31), then the next iterating point is given as $x^1 = x^0 + \alpha_0 d_{Nq}(x^0) = (1.1315, 1.3610)^T$. Clearly, we have $f(x^1) = (0.0387, 3.1327)^T < f(x^0)$. The final solution is obtained as $x^* = (1.0365, 1.0412)^T$ after 5 iterations, using the stopping criteria $|\Gamma(x^k)| < 10^{-5}$. This can also be verified that $x^*$ is an approximate weak efficient solution of $f$ by weighted sum method with weight $w = (1, 0)$.

*Generate approximate Pareto front* The multiobjective optimization problems have no single isolated minimum point but a set of efficient points. We consider a multi-start technique to generate an approximate Pareto front. A set of 100 uniformly distributed random points is collected between $lb$ and $ub$ and the proposed algorithm is executed at every initial point. The approximate Pareto front generated by Algorithm 1 is compared with the weighted sum method using the following two test problems [37]:

$$\begin{aligned} (BK1): \quad & \min \quad \left(x_1^2 + x_2^2\ ,\ (x_1-5)^2\ ,\ (x_2-5)^2\right), \\ & \text{subject to } -5 \le x_1, x_2 \le 10, \end{aligned}$$

and

$$\begin{aligned} (IM1): \quad & \min \quad \left(2\sqrt{x_1}\ ,\ x_1(1-x_2)+5\right), \\ & \text{subject to } 1 \le x_1 \le 4, \quad 1 \le x_2 \le 2. \end{aligned}$$

The single objective $q$-Newton method developed in [31] is used to solve single-objective optimization problems in the weighted sum method. We have considered

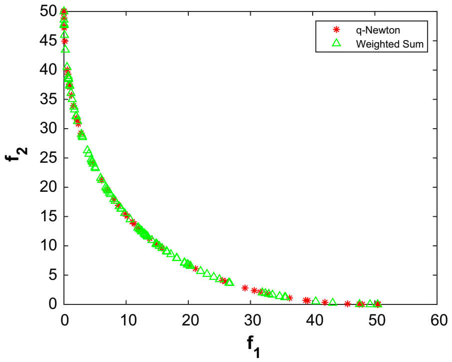


**Fig. 1** Approximate Pareto fronts of BK1 and IM1

weights $(1, 0)$, $(0, 1)$, and 98 different random positive weights. The approximate Pareto fronts of the test problems (BK1) and (IM1) are provided in Fig. 1. One can observe that Algorithm 1 provides approximate Pareto fronts for both (BK1) and (IM1). But, the weighted sum method fails to generate the approximate Pareto front in (IM1).

*Comparison with three existing schemes* Algorithm 1 (q-QN) is compared with quasi-Newton methods for multiobjective optimization problems developed in [35] (QN1), [38] (QN2), and [39] (QN3). A set of bound constrained test problems are collected from different sources, and solved using these methods. All algorithms are executed, and computational details are provided in Table 1. In this table, '*It*', '$\#F$' and '$\#G$' denote total number of iterations, function evaluations, and gradient evaluations, respectively. Total Hessian count in Algorithm 1 is equal to '*It*'. In (QN2) and (QN3), total gradient evaluations is equal to '*It*'. One can observe from Table 1 that Algorithm 1 takes less number of iterations than other methods in most cases (the lowest number of iterations are indicated by bold numbers). In view of Table 1, we can also see that ($q$-QN) has a significant improvement over (QN1), (QN2) and (QN3) relative to the number of objective function evaluations, and gradient evaluations for most of the cases. The methods (QN1), (QN2) and (QN3) update the positive definite Hessians for all $f_j$, where $j = 1, \ldots, m$, but in method ($q$-QN), we solve subproblem of (MOUP) by updating q-Hessian generated by q-derivative, which takes larger steps to get the weak efficient solutions/critical point. Hence, the q-Newton method uses better approximations of objective functions than (QN1), (QN2) and (QN3) for solving the sub-problem. Thus, from the numerical results, ($q$-QN) is superior to other existing methods presented in this paper.

## 6 Conclusion

In this paper, the $q$-calculus is used in the Newton's method for solving multiobjective unconstrained optimization problems for which existence of second order partial derivatives at every point is not required. We have given the algorithm and proved its

**Table 1** Computation details

| Problem | Source | $(m, n)$ | q-QN | | | QN1 | | | QN2 | | QN3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | $It$ | $\#F$ | $\#G$ | $It$ | $\#F$ | $\#G$ | $It$ | $\#F$ | $It$ | $\#F$ |
| BK1 | [37] | (2, 2) | **200** | 200 | 200 | **200** | 318 | 300 | 562 | 692 | **200** | 300 |
| DD1a | [40] | (2, 5) | **237** | 237 | 237 | 421 | 542 | 502 | 427 | 522 | 425 | 552 |
| DD1b | [40] | (2, 5) | **390** | 428 | 428 | 613 | 2430 | 2057 | 1757 | 1764 | 1664 | 1672 |
| DG01 | [37] | (2, 1) | **192** | 430 | 430 | 197 | 725 | 725 | 268 | 268 | 277 | 277 |
| FDSa | [32] | (3, 5) | **465** | 484 | 467 | 1712 | 3609 | 2297 | 728 | 835 | 795 | 1198 |
| FDSb | [32] | (3, 10) | **511** | 529 | 511 | 4134 | 9146 | 5428 | 1009 | 1241 | 1837 | 3771 |
| FDSc | [32] | (3, 50) | **598** | 648 | 598 | 3718 | 12968 | 5633 | 1766 | 2151 | 2339 | 3254 |
| GE5 | [41] | (3, 3) | 315 | 1503 | 1486 | **281** | 643 | 643 | 337 | 457 | 287 | 798 |
| Hil | [42] | (2, 2) | **321** | 1398 | 1076 | 334 | 1663 | 1282 | 474 | 742 | 649 | 1478 |
| IKK1 | [37] | (3, 2) | **176** | 176 | 176 | 266 | 267 | 266 | 1290 | 1920 | 232 | 232 |
| IM1 | [37] | (2, 2) | 177 | 681 | 681 | **109** | 1765 | 1765 | 310 | 310 | 311 | 311 |
| Jin1a | [43] | (2, 2 | **200** | 200 | 200 | **200** | 200 | 200 | 316 | 946 | **200** | 200 |
| Jin1b | [43] | (2, 10) | **200** | 200 | 200 | 300 | 300 | 300 | 660 | 732 | 300 | 300 |
| Jin1c | [43] | (2, 20) | **200** | 200 | 200 | 212 | 1004 | 1004 | 929 | 929 | 300 | 300 |
| KW2 | [44] | (2, 2) | **305** | 1435 | 1298 | 386 | 1560 | 1416 | 547 | 684 | 1348 | 1729 |
| lovison1 | [45] | (2, 2) | **200** | 200 | 200 | 300 | 599 | 385 | 268 | 310 | 273 | 364 |
| lovison3 | [45] | (2, 2) | 625 | 1417 | 1162 | **343** | 1683 | 1178 | 992 | 1026 | 957 | 1217 |
| lovison4 | [45] | (2, 2) | **287** | 370 | 370 | 348 | 828 | 679 | 360 | 467 | 374 | 511 |
| lovison5 | [45] | (3, 3) | 1938 | 3005 | 2675 | **803** | 2556 | 2125 | 732 | 935 | 1047 | 1416 |

**Table 1** continued

| Problem | Source | $(m, n)$ | q-QN | | | QN1 | | | QN2 | | QN3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | $It$ | $\#F$ | $\#G$ | $It$ | $\#F$ | $\#G$ | $It$ | $\#F$ | $It$ | $\#F$ |
| lovison6 | [45] | (3, 3) | 1589 | 2856 | 2468 | **835** | 4782 | 1764 | 805 | 1169 | 1305 | 3162 |
| LRS1 | [37] | (2, 2) | **200** | 200 | 200 | **200** | 300 | 300 | 4195 | 4339 | **200** | 300 |
| mhhm1 | [37] | (3, 1) | **184** | 184 | 184 | 252 | 280 | 280 | 254 | 274 | 248 | 271 |
| mhhm2 | [37] | (3, 2) | **199** | 199 | 199 | 322 | 357 | 357 | 324 | 347 | 342 | 368 |
| MLF1 | [37] | (2, 1) | 147 | 250 | 250 | **128** | 542 | 542 | 175 | 175 | 173 | 173 |
| MLF2 | [37] | (2, 2) | **170** | 1354 | 1348 | 172 | 1808 | 1782 | 1072 | 1074 | 1230 | 1286 |
| MOP1 | [37] | (2, 1) | **200** | 200 | 200 | 217 | 317 | 317 | 20000 | 20100 | 208 | 308 |
| MOP3 | [37] | (2, 2) | **311** | 803 | 746 | 344 | 786 | 704 | 454 | 509 | 500 | 1495 |
| MOP5 | [37] | (3, 2) | **104** | 315 | 209 | 118 | 119 | 118 | 199 | 415 | 160 | 160 |
| MOP7 | [37] | (3, 2) | **200** | 200 | 200 | 928 | 1079 | 1074 | 19209 | 19366 | 851 | 851 |
| PNR1 | [46] | (2, 2) | 467 | 467 | 467 | 336 | 757 | 683 | 387 | 488 | **294** | 538 |
| PNR2 | [46] | (2, 2) | **362** | 561 | 442 | 452 | 1327 | 1079 | 434 | 613 | 369 | 574 |
| PNR3 | [46] | (2, 2) | 408 | 408 | 408 | 374 | 869 | 814 | 398 | 491 | **273** | 509 |
| PNR4 | [46] | (2, 2) | **364** | 643 | 465 | 421 | 1422 | 1099 | 452 | 626 | 435 | 690 |
| PNR5 | [46] | (2, 2) | **377** | 702 | 496 | 441 | 1449 | 1077 | 450 | 601 | 474 | 731 |
| SK1 | [37] | (2, 1) | 122 | 289 | 289 | **120** | 280 | 280 | 135 | 147 | 127 | 195 |
| SK2 | [37] | (2, 4) | **543** | 1276 | 858 | 782 | 2672 | 1813 | 1014 | 1072 | 1277 | 1506 |
| SP1 | [37] | (2, 2) | **200** | 200 | 200 | 459 | 589 | 589 | 401 | 418 | 367 | 482 |
| SSFYY1 | [37] | (2, 2) | **200** | 200 | 200 | **200** | 300 | 300 | 7867 | 8083 | **200** | 300 |
| SSFYY2 | [37] | (2, 1) | **244** | 324 | 324 | 262 | 403 | 403 | 5095 | 5097 | 252 | 334 |
| VFM1 | [37] | (3, 2) | **191** | 191 | 191 | 196 | 291 | 289 | 253 | 395 | 196 | 292 |
| ZLT1 | [37] | (3, 10) | **200** | 200 | 200 | **200** | 300 | 300 | 20000 | 20100 | **200** | 300 |

convergence. The sequence provided by the method converges quadratically, and the Newton direction is chosen in the vicinity of the solution. Moreover, the quadratic convergence in case of second derivatives is Lipschitz continuous. The $q$-gradient enables the search to be carried out in a more diverse set of directions. Numerical results show that the proposed method is more efficient as compared with the other methods for solving multiobjective unconstrained optimization problems.

**Acknowledgements** We are grateful to the editors and anonymous referees for their valuable comments and detailed suggestions which helped to improve the quality of this paper. 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).

## References

1. Luc, D.T.: Theory of Vector Optimization. Lecture Notes in Economy and Mathematical Systems. Springer, New York (1989)
2. Eschenauer, H., Koski, J., Osyczka, A.: Multicriteria Design Optimization. Springer, Berlin (1990)
3. Kasperska, R., Ostwald, M., Rodak, M.: Bi-criteria optimization of open cross section of the thin-walled beams with flat flanges. Proc. Appl. Math. Mech. **4**, 614–615 (2004)
4. Jüschke, A., Jahn, J., Kirsch, A.: A bicriterial optimization problem of antenna design. Comput. Optim. Appl. **7**, 261–276 (1997)
5. Fu, Y., Diwekar, U.M.: An efficient sampling approach to multiobjective optimization. Ann. Oper. Res. **132**, 109–134 (2004)
6. Shan, S., Wang, G.G.: An efficient pareto set identification approach for multiobjective optimization on black-box functions. J. Mech. Design **127**, 866–874 (2005)
7. Carrizosa, E., Conde, E., Mumoz-Màrquez, M., Puerto, J.: Planar point-objective location problems with nonconvex constraints: a geometrical construction. J. Glob. Optim. **6**, 77–86 (1995)
8. Carrizosa, E., Frenk, J.B.G.: Dominating sets for convex functions with some applications. J. Optim. Theory Appl. **96**, 281–295 (1998)
9. Kiran, K.L., Lakshminarayanan, S.: Treatment planning of cancer dendritic cell therapy using multi-objective optimization. IFAC Proc. Vol. **42**, 109–116 (2009)
10. Petrovski, A., McCall, J., Sudha, B.: Multi-objective optimization of cancer chemotherapy using swarm intelligence. In: Taylor, N.K. (ed.) AISB Symposium on Adaptive and Emergent Behaviour and Complex Systems. UK Society for AI (2009)
11. Kiran, K.L., Jayachandran, D., Lakshminarayanan, S.: Multi-objective optimization of cancer immuno-chemotherapy. In: Lim, C.T., Goh, J.C.H. (eds.) 13th Int. Conf. Biomed. Eng., pp. 1337–1340. Springer, Singapore (2009)
12. Baesler, F.F., Sepúlveda, J.A.: Multi-objective simulation optimization for a cancer treatment center. In: Proceeding of Winter Simulation Conference, pp. 1401–1404. IEEE, USA (2001)
13. Qu, S.J., Liu, C., Goh, M.: Nonsmooth multiobjective programming with quasi-Newton methods. Eur. J. Oper. Res. **235**, 503–510 (2014)
14. Qu, S.J., Goh, M., Wu, S.Y.: Multiobjective DC programs with infinite convex constraints. J. Glob. Optim. **59**, 41–58 (2014)
15. Qu, S.J., Zhou, Y.Y., Zhang, Y.L., Wahab, M.I.M., Zhang, G., Ye, Y.Y.: Optimal strategy for a green supply chain considering shipping policy and default risk. Comput. Ind. Eng. **131**, 172–186 (2019)
16. Huang, R., Qu, S., Yang, X., Liu, Z.: Multi-stage distributionally robust optimization with risk aversion. J. Ind. Manag. Optim. (2019). https://doi.org/10.3934/jimo.2019109
17. Geoffrion, A.M.: Proper efficiency and the theory of vector maximization. J. Math. Anal. Appl. **22**, 618–630 (1968)
18. Jahn, J.: Scalarization in vector optimization. Math. Program **29**, 203–218 (1984)
19. Luc, D.T.: Scalarization of vector optimization problems. J. Optim. Theory Appl. **55**, 85–102 (1987)
20. Eichfelder, G.: Scalarizations for adaptively solving multi-objective optimization problems. J. Comput. Optim. Appl. **44**, 249–273 (2009)
21. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer Academic, Boston (1999)

22. Drummond, L.M.G., Maculan, N., Svaiter, B.: On the choice of parameters for the weighting method in vector optimization. Math. Program **111**, 201–216 (2008)
23. Jackson, F.H.: On $q$-functions and a certain difference operator. Trans. Roy. Soc. Edinburg. **46**, 253–281 (1909)
24. Ernst, T.: The history of $q$-calculus and a new method (Licentiate Thesis). U.U.D.M, Report (2000). http://www.math.uu.se/thomas/Lics.pdf
25. Jackson, F.H.: On $q$-definite integrals. Quart. J. Pure Appl. Math. **41**, 193–203 (1910)
26. Ernst, T.: A method for $q$-calculus. J. Nonlinear Math. Phys. **10**, 487–525 (2003)
27. Rajković, P.M., Marinković, S.D., Stanković, M.S.: On $q$-Newton–Kantorovich method for solving systems of equations. Appl. Math. Comput. **168**, 1432–1448 (2005)
28. Sterroni, A.C., Galski, R.L., Ramos, F.M.: The $q$-gradient vector for unconstrained continuous optimization problems. In: Hu, B., Morasch, K., Pickl, S., Siegle, M. (eds.) Oper. Res. Proc., pp. 365–370. Springer, Heidelberg (2010)
29. Gouvêa, E.J.C., Regis, R.G., Soterroni, A.C., Scarabello, M.C., Ramos, F.M.: Global optimization using $q$-gradients. Eur. J. Oper. Res. **251**, 727–738 (2016)
30. Al-Saggaf, U.M., Moinuddin, M., Arif, M., Zerguine, A.: The $q$-least mean squares algorithm. Signal Process. **111**, 50–60 (2015)
31. Chakraborty, S.K., Panda, G.: Newton like line search method using $q$-calculus. International Conference on Mathematics and Computing. In: Giri, D., Mohapatra, R.N., Begehr, H., Obaidat, M. (eds.) Communications in Computer and Information Science 655, pp. 196–208. Springer, Singapore (2017)
32. Fliege, J., Drummond, L.M.G., Svaiter, B.F.: Newton's method for multiobjective optimization. SIAM J. Optim. **20**, 602–626 (2009)
33. Drummond, L.M.G., Svaiter, B.F.: A steepest descent method for vector optimization. J. Comput. Appl. Math. **175**, 395–414 (2005)
34. Fliege, J., Svaiter, B.F.: Steepest descent methods for multicriteria optimization. Math. Methods Oper. Res. **51**, 479–494 (2000)
35. Ansary, M.A.T., Panda, G.: A modified quasi-Newton method for vector optimization problem. Optimization **64**, 2289–2306 (2015)
36. Schnabel, R.B., Eskow, E.: A revised modified cholesky factorization algorithm. SIAM J. Optim. **9**, 1135–1148 (1999)
37. Huband, S., Hingston, P., Barone, L., While, L.: A review of multiobjective test problems and a scalable test problem toolkit. IEEE T. Evolut. Comput. **10**, 477–50 (2006)
38. Qu, S., Goh, M., Chan, F.T.S.: Quasi-Newton methods for solving multiobjective optimization. Oper. Res. Lett. **39**, 397–399 (2011)
39. Povalej, Ž.: Quasi-Newton's method for multiobjective optimization. J. Comput. Appl. Math. **255**, 765–777 (2014)
40. Das, I., Dennis, J.E.: Normal-boundary intersection: a new method for generating pareto optimal points in nonlinear multicriteria optimization problems. SIAM J. Optim. **8**, 631–657 (1998)
41. Eichfelder, G.: An adaptive scalarization method in multiobjective optimization. SIAM J. Optim. **19**, 1694–1718 (2009)
42. Hillermeier, C.: Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach. Springer, Basel (2001)
43. Jin, Y., Olhofer, M., Sendhoff, B.: Dynamic weighted aggregation for evolutionary multi-objective optimization: Why does it work and how?. In: Spector, L. (ed.) Proceedings of the Genetic and Evolutionary Computation Conference. pp. 1042–1049, Morgan Kaufmann Publishers, United States (2001)
44. Kim, I.Y., de Weck, O.L.: Adaptive weighted sum method for bi-objective optimization. Struct. Multidiscip. O. **29**, 149–158 (2005)
45. Lovison, A.: A synthetic approach to multiobjective optimization (2010). arXiv:1002.0093
46. Preuss, M., Naujoks, B., Rudolph, G.: Pareto set and EMOA behavior for simple multimodal multiobjective functions. In: Runarsson, T.P., Beyer, H.G., Burke, E., Guervós, J.J.M., Whitley, L.D., Yao, X. (eds.) Parallel Problem Solving from Nature-PPSN IX, pp. 513–522. Springer, Berlin (2006)