Journal of Nonlinear and Convex Analysis
Volume 25, Number 12, 2024, 2975–2990

Yokohama Publishers
ISSN 1880-5221 ONLINE JOURNAL

A SUFFICIENT DESCENT DAI-YUAN TYPE CONJUGATE GRADIENT METHOD FOR MULTIOBJECTIVE OPTIMIZATION PROBLEMS

JIANWEN PENG*, TAO TAN, AND JEN-CHIH YAO†

Dedicated to Professor Do Sang Kim on the occasion of his 70th birthday

Abstract. In this paper, we mainly introduce a new sufficiently descended nonlinear conjugate gradient method (in short, CGM) for solving the multiobjective optimization problem (in short, MOP), which extends the modified Dai-Yuan type nonlinear CGM introduced in [X. Z. Jiang and J. B. Jian. A sufficient descent daiyuan type nonlinear conjugate gradient method for unconstrained optimization problems. Nonlinear Dynamics, 72:101-112, 2013] from the case of scalar optimization problems to that of MOPs. The novelty of this new nonlinear CGM for MOPs is that a sufficient descent direction can be always guaranteed without relying on any line search techniques. Under mild assumptions, we establish the global convergence of the proposed multiobjective CGM with Wolfe line search rule. Finally, some numerical experiments are given to illustrate the effectiveness of our proposed algorithms.

1. Introduction

In this paper, we will consider the following unconstrained multi-objective optimization problem (in short, MOP) :

$$\min_{x\in\mathbb{R}^n} F(x) = (F_1(x), F_2(x), \ldots, F_m(x))^\top, \tag{1.1}$$

where $F_i : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable for each $i \in \{1, 2, ..., m\}$ and the superscript "$\top$" denotes the transpose. Many approaches for solving (1.1) are iterative methods, and its form is given by

$$x_{k+1} = x_k + \alpha_k d_k,\ k = 0, 1, 2, \cdots ,$$

where $x_k \in \mathbb{R}^n$ is the $k-$th approximation to the solution, $d_k \in \mathbb{R}^n$ is a descent direction and $\alpha_k > 0$ is obtained by line searches. Since there is no single point which minimizes all the functions simultaneously, the concept of optimality is established in terms of Pareto optimality or efficiency. Let us recall that a point is said to be Pareto optimal or efficient, if there does not exist a different point with the same or

2020 *Mathematics Subject Classification.* 90C25, 90C31, 49K40.

Key words and phrases. Multiobjective optimization problems, conjugate gradient method, Wolfe line search, global convergence.

*J. W. Peng is supported by the National Natural Science Foundation of China (12271071, 11991024), the Basic and Advanced Research Project of Chongqing (cstc2021jcyj-msxmX0300), the Team Project of Innovation Leading Talent in Chongqing (CQYC20210309536), the Chongqing Talent Plan contract system project (cstc2022ycjh-bgzxm0147) and the Chongqing University Innovation Research Group Project (CXQT20014).

†Corresponding author. J.C. Yao is supported by the Grant MOST 113-2115-M-039-001-MY2.

smaller objective function values such that there is a decrease in at least one objective function value. Multiobjective optimization can be applied to engineering [23], statistics [3], environmental analysis [19], and management science [7], etc. Various iterative techniques exist for unconstrained multiobjective optimization problem, such as the Newton method [9, 30], quasi-Newton methods [1, 22, 27], the steepest descent method [12].

Recently, the vector versions of nonlinear conjugate gradient methods were first proposed by Lucambio Pérez and Prudente [21]. In general, the search direction in the multiobjective conjugate gradient methods can be defined by

$$d_k = \begin{cases} v(x_k), & \text{if } k = 0, \\ v(x_k) + \beta_k d_{k-1}, & \text{if } k \geq 1, \end{cases} \tag{1.2}$$

where β_k is a parameter that determines some different conjugate gradient methods, and $v(x_k)$ is the steepest descent direction in multiobjective optimization. In that paper, β_k was regarded as the extended forms of the five classical CGMs for scalar optimization parameters, namely Hentences-Stiefel (in short, HS) CGM [16], Fletcher-Reeves (in short, FR) CGM [10], Polak-Ribière-Polyak (in short, PRP) CGM [8], Conjugate descent (in short, CD) [11] and Dai-Yuan [5](in short, DY) CGM. Moreover, under mild assumptions, The global convergence of the multiobjective extensions of the Hager-Zhang and the Liu-Storey nonlinear conjugate gradient methods have been studied in [15] and [14], respectively.

Meanwhile, Lucambio Pérez and Prudente introduced the vector version of modified DY CGM in which the modified DY parameters β_k^{mDY} is given as follows:

$$\beta_k^{mDY} = \frac{-\mathcal{Q}(x_k, v(x_k))}{\mathcal{Q}(x_k, d_{k-1}) - \tau \mathcal{Q}(x_{k-1}, d_{k-1})}$$

where $\mathcal{Q}(\cdot, \cdot)$ is defined in section 2 and $\tau > 1$. It is worthy noting that if $\tau = 1$, then the above parameter becomes DY parameter $\beta_k^{DY} = \frac{-\mathcal{Q}(x_k, v(x_k))}{\mathcal{Q}(x_k, d_{k-1}) - \mathcal{Q}(x_{k-1}, d_{k-1})}$ of the nonlinear DY CGM for solving multiobjective optimization problems. However, it is worthy noting that both the sufficient descent direction at eash iteration and the global convergence of the nonlinear modified DY CGM for solving MOPs introduced in [21] can be guaranteed only on the condition that the strong Wolfe line search techniques are used. It is natural to rease the following two interesting open questions:

(i) The first one is that can we introduce a novel multioblective nonlinear CGM which can generate a sufficient descent direction without relying on any line search techniques?

(ii) The second one is that can we find any multioblective nonlinear CGM whose global convergence can be guaranteed could be guaranteed if one use the Wolfe line search step length rule instead of the strong Wolfe line search techniques?

In this paper, we will give positive answers to the above questions. On one hand, inspired by the idea of Jiang and Jian [18], we would like to introduce a novel multiobjective DY CGM in which the new modified DY parameter denoted by β_k^{NMDY} given as follows:

$$\beta_k^{NMDY} = \frac{-\mathcal{Q}(x_k, v(x_k))}{\max\{\mathcal{Q}(x_k, d_{k-1}) - \mathcal{Q}(x_{k-1}, d_{k-1}), \mu|\mathcal{Q}(x_k, d_{k-1})|\}}. \tag{1.3}$$

And we will show that the above proposed multiobjective DY type CGM can not only generate a sufficient descent direction at each iteration but also can guarantee the the global convergence of our proposed multiobjective CGM with parameter β_k^{NMDY} defined in (1.3) when the Wolfe line search step length rule is used.

The paper is organized as follows. In Section 2, we introduce some concepts, notation and preliminary results. In Section 3, we give a general scheme of the nonlinear DY type conjugate gradient method for solving MOPs. A convergence analysis of the proposed method satisfying the Wolfe conditions is performed in Section 4. In Section 5, we provide some numerical experiments to show the effectiveness of the proposed algorithm. Finally, in Section 6, we make some conclusions about our work.

2. Preliminaries

Throughtout this paper, we use $\mathbb{R}$, $\mathbb{R}_+$, and $\mathbb{R}_{++}$ stand for the sets of real numbers, non-negative real numbers, and strictly positive real numbers, respectively. Let $\Xi_m = \{1, 2, \ldots, m\}$ and $e = (1, 1, \ldots, 1)^\top \in \mathbb{R}^m$, and let $\langle \cdot, \cdot \rangle$ stand for the inner product in $\mathbb{R}^n$ and $\| \cdot \|$ denote the norm in the sense that $\|x\| = \sqrt{\langle x, x\rangle}$ for $x \in \mathbb{R}^n$. For two vectors u and v in $\mathbb{R}^m$, we use $u \preceq v$ to denote that $u_i \leq v_i$ for each $i \in \Xi_m$. Similarly, we use $u \prec v$ to denote that $u_i < v_i$ for all $i \in \Xi_m$. Given $x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$, let $F : \mathbb{R}^n \to \mathbb{R}^m$ be continuously differentiable and the Jacobian of F at x is denoted by $JF(x)$ which is defined as follows:

$$JF(x) = [\nabla F_1(x), \nabla F_2(x), \ldots, \nabla F_m(x)]^\top,$$

where $\nabla F_i(x)$ is the gradient vector of F_i at x. And the image of $JF(x)$ is denoted as

$$Im(JF(x)) = \{JF(x)d : d \in \mathbb{R}^n\}.$$

Definition 2.1 ([24])**.** A vector $x^* \in \mathbb{R}^n$ is called Pareto optimum to (MOP), if there exists no $x \in \mathbb{R}^n$ such that $F(x) \leq F(x^*)$ and $F(x) \neq F(x^*)$.

Definition 2.2 ([12])**.** A vector $x^* \in \mathbb{R}^n$ is called Pareto critical to (MOP), if

$$Im(JF(x^*)) \cap (-\mathbb{R}^m_{++}) = \emptyset,$$

where $Im(JF(x^*))$ denotes the image set of the Jacobian of F at x^*.

Definition 2.3 ([12])**.** A vector $d \in \mathbb{R}^n$ is called descent direction for F at x, if

$$JF(x)d \in -\mathbb{R}^m_{++}.$$

Now, we define $\mathcal{Q} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ as follows:

$$\mathcal{Q}(x, d) = \max_{i \in \Xi_m} \langle \nabla F_i(x), d\rangle. \tag{2.1}$$

From [4], we known that $\mathcal{Q}$ can express Pareto critical and descent direction, i.e.,

(i) $d \in \mathbb{R}^n$ is a descent direction for F at $x \in \mathbb{R}^n$ if $\mathcal{Q}(x, d) < 0$,
(ii) $x \in \mathbb{R}^n$ is Pareto critical if $\mathcal{Q}(x, d) \geq 0$ for any $d \in \mathbb{R}^n$.

The following proposition illustrates several useful results related to $\mathcal{Q}$.

Proposition 2.4 ([13])**.** *For all $x, y \in \mathbb{R}^n$, $\varrho > 0$ and $b_1, b_2 \in \mathbb{R}^n$, we obain*

(i) $\mathcal{Q}(x, \varrho b_1) = \varrho \mathcal{Q}(x, b_1)$;

(ii) $\mathcal{Q}(x, b_1 + b_2) \leq \mathcal{Q}(x, b_1) + \mathcal{Q}(x, b_2)$;
(iii) $|\mathcal{Q}(x, b_1) - \mathcal{Q}(y, b_2)| \leq \|JF(x)b_1 - JF(y)b_2\|$.

Let us now consider the following scalar optimization problem:

$$\min_{d\in\mathbb{R}^n} \mathcal{Q}(x, d) + \frac{1}{2}\|d\|^2. \tag{2.2}$$

Obviously, the objective in (2.2) is continous and strongly convex. Therefore, problem (2.2) admits a unique optimal solution. Denote the optimal solution of (2.2) by $v(x)$, i.e.,

$$v(x) = \arg\min_{d\in\mathbb{R}^n} \mathcal{Q}(x, d) + \frac{1}{2}\|d\|^2, \tag{2.3}$$

and let the optimal value of (2.2) be defined as $\theta(x)$, i.e.,

$$\theta(x) = \mathcal{Q}(x, v(x)) + \frac{1}{2}\|v(x)\|^2. \tag{2.4}$$

Observe that in scalar optimization (i.e., $m = 1$), we have $\mathcal{Q}(x, d) = \langle \nabla F_1(x), d\rangle$, $v(x) = -\nabla F_1(x)$ and $\theta(x) = -\|\nabla F_1(x)\|^2/2$.

To obtain $v(x)$, one can consider the following Lagrangian dual problem of (2.2)(see [12]):

$$\begin{aligned} \lambda(x) \in \arg\min_{\lambda\in\mathbb{R}^m} \quad & \left\|\sum_{i=1}^{m} \lambda_i \nabla F_i(x)\right\|^2 \\ s.t. \quad & \lambda \in \Lambda^m, \end{aligned} \tag{2.5}$$

where $\Lambda^m = \{\lambda \in \mathbb{R}^m : \sum_{i=1}^{m} \lambda_i = 1, \lambda_i \geq 0, \forall i \in \Xi_m\}$ stands for the simplex set. Then, $v(x)$ can also be represented as

$$v(x) = -\sum_{i=1}^{m} (\lambda(x))_i \nabla F_i(x). \tag{2.6}$$

Let us now give a characterization of Pareto critical points of problem (1.1), which will be used in our subsequent analysis.

Proposition 2.5 ([12, Lemma 1]). *Let $v(\cdot)$ and $\theta(x)$ be as in (2.3) and (2.4), respectively. Then the following statements hold:*

(i) *If x is a Pareto critical point of problem (1.1), then $v(x) = 0$ and $\theta(x) = 0$;*
(ii) *If x is not a Pareto critical point of problem (1.1), then $v(x) \neq 0, \theta(x) < 0$ and $\mathcal{Q}(x, v(x)) < -\|v(x)\|^2/2 < 0$;*
(iii) *$v(\cdot)$ is continuous.*

3. A new modification DY type conjugate gradient method for MOPs

According to the works on nonlinear conjugate gradient method (in short, CGM) for vector optimization problems in [21], we recall both the Wolfe-type line search procedure and the strong Wolfe-type line search procedure for MOPs as follows:

If $d_k \in \mathbb{R}^n$ is the direction of descent of F at x_k, given $0 < \rho < \sigma < 1$, we say that $\alpha_k > 0$ satisfies the Wolfe conditions if

$$F(x_k + \alpha_k d_k) \preceq F(x_k) + \rho \alpha_k \mathcal{Q}(x_k, d_k) e, \tag{3.1}$$

$$\mathcal{Q}(x_{k+1}, d_k) \geq \sigma \mathcal{Q}(x_k, d_k), \tag{3.2}$$

and we say that $\alpha_k > 0$ satisfies the strong Wolfe conditions if

$$F(x_k + \alpha_k d_k) \preceq F(x_k) + \rho \alpha_k \mathcal{Q}(x_k, d_k) e, \tag{3.3}$$

$$|\mathcal{Q}(x_{k+1}, d_k)| \leq \sigma |\mathcal{Q}(x_k, d_k)|. \tag{3.4}$$

Now, we propose our novel sufficient descent modified Dai-Yuan type (in short, NMDY CGM) conjugate gradient method for solving multiobjective optimization problems as follows:

Algorithm 1 (NMDY)

Step 0. Given constants $\rho \in (0,1), \sigma \in (\rho, 1), \mu > 1$, *choose initial points* $x_0 \in \mathbb{R}^n$, *let* $k \leftarrow 0$;
Step 1. Computing $\nabla F_i(x_k)$ *for each* $i \in \Xi_m$;
Step 2. Computing $\lambda(x_k)$ *and* $v(x_k)$ *by using (2.5) and (2.6),respectively;*
Step 3 If $v(x_k) = 0$ *then STOP. Otherwise, go to step 4;*
Step 4. Computing β_k^{NMDY} *by the following formula:*

$$\beta_k^{NMDY} = \frac{-\mathcal{Q}(x_k, v(x_k))}{\max\{\mathcal{Q}(x_k, d_{k-1}) - \mathcal{Q}(x_{k-1}, d_{k-1}), \mu |\mathcal{Q}(x_k, d_{k-1})|\}}; \tag{3.5}$$

Step 5. Computing d_k *by using the following formula:*

$$d_k = \begin{cases} v(x_k), & \text{if } k = 0; \\ v(x_k) + \beta_k^{NMDY} d_{k-1}, & \text{if } k \geq 1; \end{cases} \tag{3.6}$$

Step 6. Computing the step length α_k *according to the Wolfe-type line search (3.1) and (3.2);*
Step 7. Set $x_{k+1} = x_k + \alpha_k d_k$ *and let* $k := k+1$, *and go to Step 1.*

Remark 3.1. (i) It is obvious from the algorithmic framework that, when a Pareto critical point of the MOP is obtained, the NMDY CGM can successfully terminate. Thus, we assume that $v(x_k) \neq 0$ for any $k \geq 0$ in the subsequent analysis. This means that the NMDY CGM generates infinite sequences $\{x_k\}$ and $\{d_k\}$.

(ii) It is easy to verify that $0 < \beta_k^{NMDY} \leq \frac{-\mathcal{Q}(x_k, v(x_k))}{\mu |\mathcal{Q}(x_k, d_{k-1})|}$.

(iii) Algorithm 1 extends the modified DY nonlinear conjugate gradient method for solving scalar optimization introduced by Jiang and Jian [18] to the case of multiobjective optimization problems.

(iv) When $\mathcal{Q}(x_k, d_{k-1}) - \mathcal{Q}(x_{k-1}, d_{k-1}) > \mu |\mathcal{Q}(x_k, d_{k-1})|$, the β_k^{NMDY} defined in (3.5) becomes the DY-type parameter $\beta_k^{DY} = \frac{-\mathcal{Q}(x_k, v(x_k))}{\mathcal{Q}(x_k, d_{k-1}) - \mathcal{Q}(x_{k-1}, d_{k-1})}$. So, we call β_k^{NMDY} a new modified DY parameter and call Algrithm 1 as sufficient descent Dai-Yuan type conjugate gradient method for MOPs.

We recall the following well-known sufficient descent condition at x_k which is used in the convergence analysis of scalar optimization:

$$\langle \nabla F_1(x_k), d_k \rangle \leq -c \left\| \nabla F_1(x_k) \right\|^2 ,$$

where $c > 0$ is a constant.

Similarly, when analyzing convergence of the CGMs for solving vector optimization or multiobjective optimization problems, Lucambio Pérez and Prudente [21] introduced the following more stringent condition:

$$\mathcal{Q}(x_k, d_k) \leq c\mathcal{Q}(x_k, v(x_k)), \tag{3.7}$$

which means that the direction d_k satisfies the sufficient descent condition at x_k.

The following lemma shows that the search direction in Algorithm 1 defined by (3.5) and (3.6) is always sufficient descent.

Lemma 3.2. *Let $\{x_k, d_k\}$ be the sequence generated by Algorithm 1 and $\mu > 1$. Then for any $k \geq 0$, d_k satisfies the sufficient descent condition (3.7) with $c = 1 - \frac{1}{\mu}$, i.e.*

$$\mathcal{Q}(x_k, d_k) \leq \left(1 - \frac{1}{\mu}\right) \mathcal{Q}(x_k, v(x_k)).$$

Proof. Case 1: if $k = 0$. By Proposition 2.5(ii), we obtain that $\mathcal{Q}(x_0, v(x_0)) < 0$. It follows from (3.6) and $\mu > 1$ that

$$\mathcal{Q}(x_0, d_0) = \mathcal{Q}(x_0, v(x_0)) \leq \left(1 - \frac{1}{\mu}\right) \mathcal{Q}(x_0, v(x_0)).$$

Case 2: $k \geq 1$. From Remark 3.1(ii), we know that $\beta_k^{NMDY} > 0$. It can be inferred from (3.6) and Proposition 2.4 that,

$$\mathcal{Q}(x_k, d_k) \leq \mathcal{Q}(x_k, v(x_k)) + \beta_k^{NMDY} \mathcal{Q}(x_k, d_{k-1}). \tag{3.8}$$

On one hand, if $\mathcal{Q}(x_k, d_{k-1}) \leq 0$, it can be derived from Remark 3.1(i) and Proposition 2.5(ii) that $\mathcal{Q}(x_k, v(x_k)) < 0$. Thus, by (3.8), $\mu > 1$ and $\beta_k^{NMDY} > 0$, we have

$$\mathcal{Q}(x_k, d_k) \leq \mathcal{Q}(x_k, v(x_k)) \leq \left(1 - \frac{1}{\mu}\right) \mathcal{Q}(x_k, v(x_k)).$$

On the other hand, if $\mathcal{Q}(x_k, d_{k-1}) > 0$, by Remark 3.1(ii) and (3.8), we have

$$\begin{aligned}
\mathcal{Q}(x_k, d_k) &\leq \mathcal{Q}(x_k, v(x_k)) + \beta_k^{NMDY} \mathcal{Q}(x_k, d_{k-1}) \\
&\leq \mathcal{Q}(x_k, v(x_k)) + \frac{-\mathcal{Q}(x_k, v(x_k))}{\mu \mathcal{Q}(x_k, d_{k-1})} \mathcal{Q}(x_k, d_{k-1}) \\
&= \mathcal{Q}(x_k, v(x_k)) - \frac{\mathcal{Q}(x_k, v(x_k))}{\mu} \\
&= \left(1 - \frac{1}{\mu}\right) \mathcal{Q}(x_k, v(x_k)).
\end{aligned}$$

Hence, the assert follows from two inequalities and the proof is complete. $\square$

Lemma 3.3. *Let $\{x_k, d_k\}$ be the sequence generated by Algorithm 1. Then for any $k \geq 1$, we have*

$$0 < \beta_k^{NMDY} \leq \frac{\mathcal{Q}(x_k, d_k)}{\mathcal{Q}(x_{k-1}, d_{k-1})}.$$

Proof. Case 1: $\mathcal{Q}(x_k, d_{k-1}) = 0$. It follows from (3.5) that

$$0 < \beta_k^{NMDY} = \frac{-\mathcal{Q}(x_k, v(x_k))}{-\mathcal{Q}(x_{k-1}, d_{k-1})} = \frac{\mathcal{Q}(x_k, v(x_k))}{\mathcal{Q}(x_{k-1}, d_{k-1})}. \tag{3.9}$$

It follows from (3.8) that

$$\mathcal{Q}(x_k, d_k) \leq \mathcal{Q}(x_k, v(x_k)) < 0.$$

By Lemma 3.2, we obtain $\mathcal{Q}(x_0, d_0) < 0$. Thus, from (3.9) and the above inequality that

$$\frac{\mathcal{Q}(x_k, d_k)}{\mathcal{Q}(x_{k-1}, d_{k-1})} \geq \frac{\mathcal{Q}(x_k, v(x_k))}{\mathcal{Q}(x_{k-1}, d_{k-1})} = \beta_k^{NMDY} > 0$$

Case 2: $\mathcal{Q}(x_k, d_{k-1}) \neq 0$ and $\mathcal{Q}(x_k, d_{k-1}) - \mathcal{Q}(x_{k-1}, d_{k-1}) > \mu|\mathcal{Q}(x_k, d_{k-1})|$. From (3.5), one has

$$\beta_k^{NMDY} = \frac{-\mathcal{Q}(x_k, v(x_k))}{\mathcal{Q}(x_k, d_{k-1}) - \mathcal{Q}(x_{k-1}, d_{k-1})}.$$

Substituting the above equation into (3.8), we obtain

$$\begin{aligned}
\mathcal{Q}(x_k, d_k) &\leq \mathcal{Q}(x_k, v(x_k)) + \beta_k^{NMDY} \mathcal{Q}(x_k, d_{k-1}) \\
&= \mathcal{Q}(x_k, v(x_k)) + \frac{-\mathcal{Q}(x_k, v(x_k))\mathcal{Q}(x_k, d_{k-1})}{\mathcal{Q}(x_k, d_{k-1}) - \mathcal{Q}(x_{k-1}, d_{k-1})} \\
&= \frac{-\mathcal{Q}(x_k, v(x_k))\mathcal{Q}(x_{k-1}, d_{k-1})}{\mathcal{Q}(x_k, d_{k-1}) - \mathcal{Q}(x_{k-1}, d_{k-1})} \\
&= \beta_k^{NMDY} \mathcal{Q}(x_{k-1}, d_{k-1}).
\end{aligned}$$

It follows from the above inequality and $\mathcal{Q}(x_{k-1}, d_{k-1}) < 0$ that $\beta_k^{NMDY} \leq \frac{\mathcal{Q}(x_k, d_k)}{\mathcal{Q}(x_{k-1}, d_{k-1})}$.

Case 3: $\mathcal{Q}(x_k, d_{k-1}) \neq 0$ and $\mathcal{Q}(x_k, d_{k-1}) - \mathcal{Q}(x_{k-1}, d_{k-1}) \leq \mu|\mathcal{Q}(x_k, d_{k-1})|$. On one hand, if $\mathcal{Q}(x_k, d_{k-1}) > 0$, then the inequality $\mathcal{Q}(x_k, d_{k-1}) - \mathcal{Q}(x_{k-1}, d_{k-1}) \leq \mu|\mathcal{Q}(x_k, d_{k-1})|$ can be rewritten as

$$(1 - \mu)\mathcal{Q}(x_k, d_{k-1}) \leq \mathcal{Q}(x_{k-1}, d_{k-1}). \tag{3.10}$$

From (3.5), we get $\beta_k^{NMDY} = \frac{-\mathcal{Q}(x_k, v(x_k))}{\mu \mathcal{Q}(x_k, d_{k-1})}$. According to (3.10) and (3.8), we have

$$\begin{aligned}
\mathcal{Q}(x_k, d_k) &\leq \mathcal{Q}(x_k, v(x_k)) + \beta_k^{NMDY} \mathcal{Q}(x_k, d_{k-1}) \\
&= \mathcal{Q}(x_k, v(x_k)) + \frac{-\mathcal{Q}(x_k, v(x_k))\mathcal{Q}(x_k, d_{k-1})}{\mu \mathcal{Q}(x_k, d_{k-1})} \\
&= \frac{-(1-\mu)\mathcal{Q}(x_k, v(x_k))\mathcal{Q}(x_k, d_{k-1})}{\mu \mathcal{Q}(x_k, d_{k-1})} \\
&\leq \frac{-\mathcal{Q}(x_k, v(x_k))\mathcal{Q}(x_{k-1}, d_{k-1})}{\mu \mathcal{Q}(x_k, d_{k-1})} \\
&= \beta_k^{NMDY} \mathcal{Q}(x_{k-1}, d_{k-1}),
\end{aligned}$$

which implies that $\beta_k^{NMDY} \leq \frac{\mathcal{Q}(x_k, d_k)}{\mathcal{Q}(x_{k-1}, d_{k-1})}$.

On the other hand, if $\mathcal{Q}(x_k, d_{k-1}) < 0$, then $\mathcal{Q}(x_k, d_{k-1}) - \mathcal{Q}(x_{k-1}, d_{k-1}) \leq \mu|\mathcal{Q}(x_k, d_{k-1})|$ can be rewritten as $(1+\mu)\mathcal{Q}(x_k, d_{k-1}) \leq \mathcal{Q}(x_{k-1}, d_{k-1})$. From (3.5), we obtain $\beta_k^{NMDY} = \frac{-\mathcal{Q}(x_k, v(x_k))}{-\mu \mathcal{Q}(x_k, d_{k-1})}$. Furthermore, one has

$$\begin{aligned}
\mathcal{Q}(x_k, d_k) &\leq \mathcal{Q}(x_k, v(x_k)) + \beta_k^{NMDY} \mathcal{Q}(x_k, d_{k-1}) \\
&= \mathcal{Q}(x_k, v(x_k)) + \frac{-\mathcal{Q}(x_k, v(x_k))\mathcal{Q}(x_k, d_{k-1})}{-\mu \mathcal{Q}(x_k, d_{k-1})} \\
&= \frac{(1+\mu)\mathcal{Q}(x_k, v(x_k))\mathcal{Q}(x_k, d_{k-1})}{\mu \mathcal{Q}(x_k, d_{k-1})} \\
&\leq \frac{\mathcal{Q}(x_k, v(x_k))\mathcal{Q}(x_{k-1}, d_{k-1})}{\mu \mathcal{Q}(x_k, d_{k-1})} \\
&= \beta_k^{NMDY} \mathcal{Q}(x_{k-1}, d_{k-1}),
\end{aligned}$$

which implies that $\beta_k^{NMDY} \leq \frac{\mathcal{Q}(x_k, d_k)}{\mathcal{Q}(x_{k-1}, d_{k-1})}$. Therefore the proof is complete. □

4. Convergence analysis

In this section, we show the global convergence property of the proposed NMDY CGM for MOPs. For this purpose, we need the following conditions:

Condition A1. The mapping $F : \mathbb{R}^n \to \mathbb{R}^m$ is bounded from below on the level set $\mathcal{L}(x_0) = \{x \in \mathbb{R}^n : F(x) \preceq F(x_0)\}$, where $x_0 \in \mathbb{R}^n$ is an available point and the set $\mathcal{L}(x_0)$ is bounded.

Condition A2. The Jacobian JF is Lipschitz continuous in an open convex set $\mathcal{B}$ that contains $\mathcal{L}(x_0)$, i.e., there exists a constant $L > 0$ such that $\|JF(x) - JF(y)\| \leq L\|x - y\|$ for all $x, y \in \mathcal{B}$.

Lemma 4.1. *Assume that Condition A2 holds true and the sequence $\{x_k, d_k\}$ is generated by Algorithm 1. Then there is a positive constant ω satisfying*

$$F(x_k) - F(x_{k+1}) \succeq \omega \frac{\mathcal{Q}^2(x_k, d_k)}{||d_k||^2}, \ \textit{for any } k, \tag{4.1}$$

where $\omega = \frac{\rho(1-\sigma)}{L}$.

Proof. From Lemma 3.2, we obtain that $\mathcal{Q}(x_k, d_k) \leq c\mathcal{Q}(x_k, v(x_k)) < 0$. It follows from (3.2) that

$$(\sigma - 1)\mathcal{Q}(x_k, d_k) \leq \mathcal{Q}(x_{k+1}, d_k) - \mathcal{Q}(x_k, d_k). \tag{4.2}$$

From Proposition 2.4(iii), (A2) and Cauchy-Schwartz inequality, we get

$$\begin{aligned}\mathcal{Q}(x_{k+1}, d_k) - \mathcal{Q}(x_k, d_k) &\leq \|JF(x_{k+1})d_k - JF(x_k)d_k\| \\ &\leq \|JF(x_{k+1}) - JF(x_k)\|\|d_k\| \\ &\leq L\alpha_k\|d_k\|^2,\end{aligned}$$

which combined with (4.2) and the fact that $\|d_k\| \neq 0$ gives

$$\alpha_k \geq \frac{\sigma - 1}{L}\frac{\mathcal{Q}(x_k, d_k)}{\|d_k\|^2} > 0. \tag{4.3}$$

By (3.1) and (4.3), it follows that for each $i \in \Xi_m$,

$$F_i(x_k) - F_i(x_{k+1}) \geq -\rho\alpha_k\mathcal{Q}(x_k, d_k) \geq \omega\frac{\mathcal{Q}^2(x_k, d_k)}{\|d_k\|^2}, \tag{4.4}$$

where $\omega = \frac{\rho(1-\sigma)}{L}$, which concludes the proof. □

Remark 4.2. Based on Condition A1 and Lemma 4.1, $\{x_k\}$ is contained in the bounded set $\mathcal{L}(x_0)$. By Proposition 2.5(iii), we have $\{v(x_k)\}$ and $\{d_k\}$ is bounded. That is, there exist constants $\xi_1, \xi_2 > 0$ such that $\|v(x_k)\| \leq \xi_1$ and $\|d_k\| \leq \xi_2$.

According to (4.1) and Condition A1, we know that $\{F(x_k)\}$ is monotone non-increasing and bounded below for $k \geq 0$, hence it is convergent. Then, we can easily obtain the following result:

Lemma 4.3. *Let Condition A1 and Condition A2 hold true, and the sequence* $\{x_k, d_k\}$ *is generated by Algorithm 1. Then*

$$\sum_{k=0}^{\infty}\frac{\mathcal{Q}^2(x_k, d_k)}{\|d_k\|^2} < +\infty. \tag{4.5}$$

We now establish the global convergence of Algorithm 1.

Theorem 4.4. *Suppose that Condition A2 holds true and the sequence* $\{x_k, d_k\}$ *is generated by the Algorithm 1. Then*

$$\lim_{k\to\infty}\inf\|v(x_k)\| = 0.$$

Proof. Suppose by contradiction that there is $\varsigma > 0$ and a positive index $\mathbf{N}$ such that

$$\|v(x_k)\| \geq \varsigma$$

for any $k \geq \mathbf{N}$.
It follows from (3.6) and Lemma 3.3 that for each $k \in \mathbf{N}$

$$\begin{aligned}\|d_k\| &\leq \|v(x_k)\| + \|\beta_k^{NMDY}d_{k-1}\| \\ &\leq \|v(x_k)\| + \frac{\mathcal{Q}(x_k, d_k)\|d_{k-1}\|}{\mathcal{Q}(x_{k-1}, d_{k-1})}.\end{aligned} \tag{4.6}$$

By Proposition 2.5(ii), we get $\theta(x_k) = \mathcal{Q}(x_k, v(x_k)) + \frac{1}{2}\|v(x_k)\|^2 < 0$. Thus, From Lemma 3.2, we obtain that for all $k \geq \mathbf{N}$,

$$0 < \varsigma^2 \leq \|v(x_k)\|^2 < -2\mathcal{Q}(x_k, v(x_k)) \leq -2\frac{\mathcal{Q}(x_k, d_k)}{c}. \tag{4.7}$$

By (4.6) and (4.7), one has

$$\begin{aligned}\frac{\|d_k\|}{\mathcal{Q}(x_k, d_k)} &\geq \frac{\|v(x_k)\|}{\mathcal{Q}(x_k, d_k)} + \frac{\|d_{k-1}\|}{\mathcal{Q}(x_{k-1}, d_{k-1})} \\ &\geq \frac{\|v(x_k)\|}{c\mathcal{Q}(x_k, v(x_k))} + \frac{\|d_{k-1}\|}{\mathcal{Q}(x_{k-1}, d_{k-1})} \\ &> -\frac{2}{c\|v(x_k)\|} + \frac{\|d_{k-1}\|}{\mathcal{Q}(x_{k-1}, d_{k-1})}.\end{aligned}$$

Then by the above inequality, we have

$$\frac{\|d_k\|}{\mathcal{Q}(x_k, d_k)} - \frac{\|d_{k-1}\|}{\mathcal{Q}(x_{k-1}, d_{k-1})} > -\frac{2}{c\|v(x_k)\|} \geq -\frac{2}{c\varsigma}, \tag{4.8}$$

which implies that for all $k \geq \mathbf{N}$,

$$\frac{\|d_k\|}{\mathcal{Q}(x_k, d_k)} - \frac{\|d_{\mathbf{N}-1}\|}{\mathcal{Q}(x_{\mathbf{N}-1}, d_{\mathbf{N}-1})} > -\frac{2(k-\mathbf{N}+1)}{c\varsigma}. \tag{4.9}$$

From Lemma 3.2, Proposition 2.5(ii) and Remark 4.2, we have

$$\frac{\|d_{\mathbf{N}-1}\|}{\mathcal{Q}(x_{\mathbf{N}-1}, d_{\mathbf{N}-1})} \geq \frac{\|d_{\mathbf{N}-1}\|}{c\mathcal{Q}(x_{\mathbf{N}-1}, v(x_{\mathbf{N}-1}))} > -\frac{\|d_{\mathbf{N}-1}\|}{c\|v(x_{\mathbf{N}-1})\|^2} \geq -\frac{\xi_1}{c\xi_2^2}, \tag{4.10}$$

It follows from (4.9) and (4.10) that for all $k \geq \mathbf{N}$,

$$\begin{aligned}\frac{\|d_k\|}{\mathcal{Q}(x_k, d_k)} &> \frac{\|d_{\mathbf{N}-1}\|}{\mathcal{Q}(x_{\mathbf{N}-1}, d_{\mathbf{N}-1})} - \frac{2(k-\mathbf{N}+1)}{c\varsigma} \\ &> -\frac{\xi_1}{c\xi_2^2} - \frac{2(k-\mathbf{N}+1)}{c\varsigma} \\ &= -\frac{\xi_1\varsigma + 2(k-\mathbf{N}+1)\xi_2^2}{c\varsigma\xi_2^2}.\end{aligned}$$

For convenience, we let $\vartheta = -\frac{\xi_1\varsigma+2(k-\mathbf{N}+1)\xi_2^2}{c\varsigma\xi_2^2}$. Therefore,

$$\frac{\|d_k\|^2}{\mathcal{Q}^2(x_k, d_k)} < \vartheta^2,$$

which implies that

$$\frac{\mathcal{Q}^2(x_k, d_k)}{\|d_k\|^2} \geq \frac{1}{\vartheta^2}.$$

By summing up the above inequality from $k = \mathbf{N}$ to infinity, we obtain

$$\sum_{k=\mathbf{N}}^{\infty} \frac{\mathcal{Q}^2(x_k, d_k)}{\|d_k\|^2} = \infty,$$

which contradicts to (4.5) in Lemma 4.3. This completes the proof. □

5. Numerical experiments

In this section, we present some numerical results and demonstrate the numerical performance of Algorithm 1 for different problems and just compare Algorithm 1 (NMDY) with the multi-objective versions of the Polak-Ribire-Polyak (PRP), Dai-Yuan (DY) and the modified DY(mDY) nonlinear conjugate gradient methods proposed in [21]. The code for the Algorithm 1 was written in double-precision Fortran 90 and was run using cmd on a PC equipped with a 3.40 GHz CPU and 8 GB RAM. For computing the steepest descent direction $v(x)$, we use Algencan [2], which is an augmented Lagrangian code for general nonlinear programming.

In our implementation, we choose the step length α_k to satisfy the Wolfe conditions for Algorithm 1 and we take step-size α_k that satisfies the strong Wolfe condition for the other three methods with $\rho = 10^{-4}$ and $\sigma = 10^{-1}$. We employ $\mu = 11.75$ for our algorithm, which seemed to work reasonably well for a broad class of problems.

Since Proposition 2.5 implies that $v(x) = 0$ if and only if $\theta(x) = 0$, we stop the algorithm at x_k whenever

$$\theta(x_k) \geq -5 \times eps^{1/2},$$

where $\theta(x_k) = \mathcal{Q}(x_k, v(x_k)) + \|v(x_k)\|^2/2$, and $eps = 2^{-52} \approx 2.22 \times 10^{-16}$ denotes the machine precision. The maximum number of allowed iterations is set to 10000. The latter stopping criterion corresponds to the fault.

The test problems considered have been collected from the literature as described in Table 1. The first two columns identify the name of the problem and the corresponding source. The starting points belonging to the box constraint $\{x \in \mathbb{R}^n | x_L \leq x \leq x_U\}$ are reported in the third and fourth columns of the table, respectively, where the lower bound $x_L \in \mathbb{R}^n$ and upper bound $x_U \in \mathbb{R}^n$. The "n" and "m" in the fifth and sixth columns denote the number of variables and the number of objective functions of the test problem, respectively. "Convex" informs whether the problem is convex or not.

We use the performance profile proposed by Dolan and Moré in [6] to compare the performance of the algorithms. Let size A denote the number of elements of the set A, $\mathcal{S}$ be the set of solvers, $\mathcal{P}$ be the set of problems, and $t_{p,s}$ be the performance of the solver $s \in \mathcal{S}$ on the problem $p \in \mathcal{P}$. The performance ratio is $r_{p,s} := \frac{t_{p,s}}{\min\{t_{p,s}: s \in \mathcal{S}\}}$ and the cumulative distribution function $\rho_s : [1, \infty) \to [0, 1]$ is

$$\rho_s := \frac{1}{n_p} \text{size}\{p \in \mathcal{P} : r_{p,s} \leq \tau\}.$$

Therefore, the performance profile is presented by depicting the graph of the cumulative distribution function ρ_s. Note that $\rho_s(1)$ is the probability that the solver beats the remaining solvers.

We take into account 200 beginning points from a uniform random distribution that belonged to the matching boxes for each method. Every instance is regarded as a separate problem, and each solver that is taken into consideration to solve it. The CPU time (T cpu), number of iterations (Itr), number of function evaluations (NF), and number of gradient evaluations (NG) are the performance metrics we use to compare the methods.

Table 1. List of test problems

Problem	Soure	x_L	x_U	n	m	Convex
JOS1	[17]	-[10000,$\cdots$,10000]	[10000,$\cdots$,10000]	1000	2	Y
SLC2	[28]	-[100,$\cdots$,100]	[100,$\cdots$,100]	100	2	N
SLCDT1	[29]	-[5,5]	[5,5]	2	2	N
AP2	[1]	-[100,100,100]	[100,100,100]	3	3	Y
AP3	[1]	-[100,100]	[100,100]	2	2	N
Lov1	[20]	-[100,100]	[100,100]	2	2	Y
Lov3	[20]	-[100,100]	[100,100]	2	2	N
Lov4	[20]	-[100,100]	[100,100]	2	2	N
FDS	[9]	-[2,$\cdots$,2]	[2,$\cdots$,2]	50	3	Y
MMR5	[25]	-[5,$\cdots$,5]	[5,$\cdots$,5]	100	2	N
MGH16	[26]	-[100,$\cdots$,100]	[100,$\cdots$,100]	4	100	N
MOP2	[17]	-[1,1]	[1,1]	2	2	N
MOP7	[17]	-[400,400]	[400,400]	2	3	Y
DGO1	[17]	-10	13	1	2	Y
Far1	[17]	-[1,1]	[1,1]	2	2	N
MLF2	[17]	[0,0]	[20,20]	2	2	Y

Table 2. Performance of the NMDY, and DY parameter on the chosen set of test problems

Problem	NMDY				DY			
	iteration	evalf	evalg	time	iteration	evalf	evalg	time
JOS1	2	2	4	0.02	2	2	4	0.05
SLC2	24	178	150	0.33	33	295	252	0.38
SLCDT1	4	18	17	0.02	3	13	13	0.02
AP2	2	2	4	0.02	2	2	4	0.02
AP3	28	216	181	0.02	116	641	634	0.05
Lov1	3	4	6	0.02	4	6	8	0.02
Lov3	4	9	11	0.02	4	9	11	0.02
Lov4	2	3	5	0.05	2	3	5	0.02
FDS	127	968	575	9.95	496	3647	3478	34.2
MMR5	37	303	292	0.3	141	744	712	0.61
MGH16	7	1309	1405	0.02	24	6439	5293	0.03
MOP2	9	62	51	0.02	7	54	48	0.02
MOP7	7	18	21	0.02	9	24	27	0.02
DGO1	2	7	9	0.02	2	7	9	0.02
FAR1	173	884	794	0.05	717	3201	3199	0.03
MLF2	28	197	157	0.02	30	221	212	0.02

TABLE 3. Performance of the PRP and mDY parameter on the chosen set of test problems

Problem	PRP				mDY			
	iteration	evalf	evalg	time	iteration	evalf	evalg	time
JOS1	2	2	4	0.03	2	2	4	0.05
SLC2	56	312	292	0.42	166	1180	856	0.38
SLCDT1	4	17	18	0.02	3	13	13	0.02
AP2	2	4	2	0.02	2	2	4	0.05
AP3	36	197	185	0.03	907	6359	4560	0.05
Lov1	3	4	6	0.02	4	6	8	0.02
Lov3	4	9	11	0.02	4	9	11	0.02
Lov4	2	3	5	0.05	2	3	5	0.02
FDS	78	532	491	5.22	4854	43711	34038	258.61
MMR5	27	218	214	0.23	1189	5106	5044	2.91
MGH16	13	2534	2635	0.02	9	2242	2033	0.02
MOP2	11	56	61	0.02	7	49	44	0.02
MOP7	6	15	18	0.02	7	18	21	0.02
DGO1	2	7	9	0.02	2	7	9	0.02
FAR1	135	724	646	0.05	7047	49323	35265	0.42
MLF2	42	234	206	0.03	157	1109	834	0.06

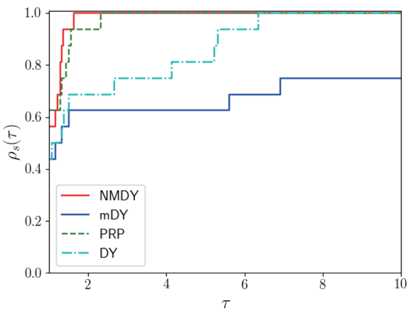

FIGURE 1. Performance profile for the Itr

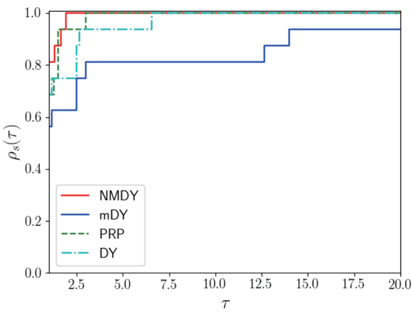

FIGURE 2. Performance profile for the T cpu

As can be seen from Tables 2 and 3 and Figures 1 to 4, our proposed Algorithm 1 is slightly better than the PRP conjugate gradient method in the four indicators examined, and is significantly higher than the mDY and DY conjugate gradient methods. Meanwhile, the mDY conjugate gradient method is the least effective among the four indicators evaluated. That's because Lucambio Pérez and Prudente only made simple modification to the DY parameters to achieve global convergence under strong Wolfe conditions and did not consider the improvement of algorithm performance. It can be seen that our modification of the DY parameter has greatly

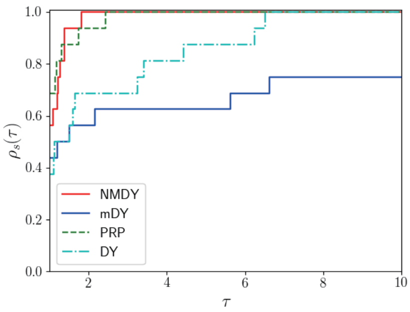

FIGURE 3. Performance profile for the NF

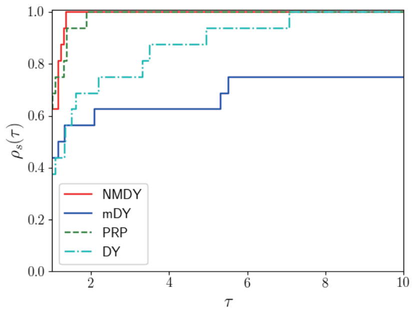

FIGURE 4. Performance profile for the NG

improved the performance of the algorithm, which verifies the effectiveness of our algorithm.

6. Conclusions

In this work, we present a new nonlinear modified Dai-Yuan type conjugate gradient method for solving unconstrained MOPs by modifying the parameter β_k^{DY} replaced by β_k^{NMDY}. The proposed multiobjective conjugate gradient method not only extends the conjugate gradient method introduced by Jiang and Jian from the scalar optimization case to that of multiobjective optimization, but also improve the Dai-Yuan type conjugate gradient method for solving MOPs. The advantage of the proposed multiobjective conjugate gradient method lies in that it can generate a sufficient descent direction without relying on any line search conditions. Under mild assumptions, we establish the global convergence of our nonlinear conjugate gradient method for MOPs with the standard Wolfe line search instead of the strong Wolfe line search. Numerical results demonstrate the validity of the proposed method.

References

[1] M. A. T. Ansary and G. Panda, *A modified quasi-newton method for vector optimization problem*, Optimization. **64** (2015), 2289–2306.

[2] E. G. Birgin and J. M. Martínez, *Practical Augmented Lagrangian Methods for Constrained Optimization*, Fundam. Algorithms 10, SIAM, Philadelphia, PA, 2014.

[3] E. Carrizosa and J. B. G. Frenk, *Dominating sets for convex functions with some applications*, J. Optim Theory Appl. **90** (1998), 281–295

[4] W. Chen, X. M. Yang and Y. Zhao, *Memory gradient method for multiobjective optimization*, Appl. Math. Comput. **443** (2023), 127791.

[5] Y. H. Dai and Y. X. Yuan, *A nonlinear conjugate gradient method with a strong global convergence property*, SIAM J. Optim. **10** (1999), 177–182.

[6] E. D. Dolan and J. J. Moré, *Benchmarking optimization software with performance profiles*, Math. Program. **91** (2001), 201–213.

[7] G. W. Evans, *An overview of techniques for solving multiobjective mathematical programs*, Manage. Sci. **30** (1984), 1268–1282.

[8] E. Polak and G. Ribière, *Note sur la convergence de méthodes de directions conjuguées*, Rev. Francaise Inform. Rech. Opér. Sér. Rouge. **3** (1969), 35–43.

[9] J. Fliege, L. M. Graña Drummond and B. F. Svaiter, *Newton's method for multiobjective optimization*, SIAM J. Optim. **20** (2009), 602–626.

[10] R. Fletcher and C. M. Reeves, *Function minimization by conjugate gradients*, Comput. J. **7** (1964), 149–154.

[11] R. Fletcher, *Part I: Unconstrained optimization*, in: Pract. Methods Optim., John Wiley, NewYork, 1980.

[12] J. Fliege and B. F. Svaiter, *Steepest descent methods for multicriteria optimization*, Math. Methods Oper. Res. **51** (2000), 479–494.

[13] E. H. Fukuda and L. M. Graña Drummond, *A survey on multiobjective descent methods.* Pesquisa Oper. **34** (2014), 585–620.

[14] M. L. N. Goncalves, F. S. Lima and L. F. Prudente, *A study of Liu-Storey conjugate gradient methods for vector optimization*, Appl. Math. Comput. **425** (2022): 127099.

[15] M. L. N. Goncalves and L. F. Prudente, *On the extension of the Hager-Zhang conjugate gradient method for vector optimization*, Comput. Optim. Appl. **76** (2020), 889–916.

[16] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Stand. **49** (1952), 409–436.

[17] S. Huband, P. Hingston, L. Barone and L. While, *A review of multiobjective test problems and a scalable test problem toolkit*, IEEE Trans. Evol. Comput. **10** (2006), 477–506.

[18] X. Z. Jiang and J. B. Jian, *A sufficient descent daiyuan type nonlinear conjugate gradient method for unconstrained optimization problems*, Nonlinear Dyn. **72** (2013), 101–112.

[19] T. M. Leschine, H. Wallenius and W. A. Verdini, *Interactive multiobjective analysis and assimilative capacity-based ocean disposal decision*, Eur. J. Oper. Res. **56** (1992), 278–289.

[20] A. Lovison, *Singular continuation: Generating piecewise linear approximations to Pareto sets via globalanalysis*, SIAM J.Optim. **21** (2011), 463–490.

[21] L. R. Lucambio Pérez and L. F. Prudente, *Nonlinear conjugate gradient methods for vector optimization*, SIAM J. Optim. **28** (2018), 2690–2720.

[22] N. Mahdavi-Amiri and F. Salehi Sadaghiani, *A superlinearly convergent nonmonotone quasi-newton method for unconstrained multiobjective optimization*, Optim Meth. and Softw. **35** (2020), 1223–1247.

[23] R. T. Marler and J. S. Arora, *Survey of multi-objective optimization methods for engineering*, Struct. Multidiscip. Optim. **26** (2004), 369–395.

[24] K. Miettinen, *Nonlinear Multiobjective Optimization*, vol. 12, Springer Science and Business Media, 1999.

[25] E. Miglierina, E. Molho and M. C. Recchioni, *Box-constrained multi-objective optimization: A gradient-like method without a priori scalarization*, Eur. J. Oper. Res. **188** (2008), 662–682.

[26] J. J. Moré, B. S. Garbow and K. E. Hillstrom, *Testing unconstrained optimization software*, ACM Trans. Math. Software. **7** (1981), 17–41.

[27] S. J. Qu, M. Goh and F. T. S. Chan, *Quasi-Newton methods for solving multiobjective optimization.* Oper. Res. Lett. **39** (2011), 397–399.

[28] O. Schütze, A. Lara and C. A. Coello, *The directed search method for unconstrained multi-objective optimization problems*, Technical report TR-OS-2010-01, http://delta.cs.cinvestav.mx/schuetze/technicalreports/TR-OS-2010-01.pdf.gz.

[29] O. Schütze, M. Laumanns, C. A. Coello, M. Dellnitz and E.-G. Talbi, *Convergence of stochastic search algorithms to finite size pareto set approximations*, J. Global Optim. **41** (2008), 559–557.

[30] J. Wang, Y. Hu, C. K. W. Yu, C. Li and X. Yang, *Extended newton methods for multiob-jective optimization: Majorizing function technique and convergence analysis*, SIAM J. Optim. **29** (2019), 2388–2421.

Manuscript received January 11, 2024
revised May 21, 2024

J. W. Peng
School of Mathematical Sciences, Chongqing Normal University, Chongqing 401331, P. R. China
E-mail address: jwpeng168@hotmail.com

T. Tan
School of Mathematical Sciences, Chongqing Normal University, Chongqing 401331, P. R. China
E-mail address: 1923806974@qq.com

J. C. Yao
Research Center for Interneural Computing, China Medical University, Taichung 40402, Taiwan and Department of Applied Mathematics, National Sun Yat-sen University, Kaohsiung 804, Taiwan
E-mail address: yaojc@mail.cmu.edu.tw