Conditional gradient method for vector optimization

Wang Chen · Xinmin Yang · Yong Zhao

Abstract

In this paper, we propose a conditional gradient method for solving constrained vector optimization problems with respect to a partial order induced by a closed, convex and pointed cone with nonempty interior. When the partial order under consideration is the one induced by the non-negative orthant, we regain the method for multiobjective optimization recently proposed by Assunção et al. (Comput Optim Appl 78(3):741–768, 2021). In our method, the construction of the auxiliary subproblem is based on the well-known oriented distance function. Three different types of step size strategies (Armijo, adaptive and nonmonotone) are considered. Without convexity assumption related to the objective function, we obtain the stationarity of accumulation points of the sequences produced by the proposed method equipped with the Armijo or the nonmonotone step size rule. To obtain the convergence result of the method with the adaptive step size strategy, we introduce a useful cone convexity condition which allows us to circumvent the intricate question of the Lipschitz continuity of Jacobian for the objective function. This condition helps us to generalize the classical descent lemma to the vector optimization case. Under convexity assumption for the objective function, it is proved that all accumulation points of any generated sequences obtained by our method are weakly efficient solutions. Numerical experiments illustrating the practical behavior of the methods are presented.

Xinmin Yang
xmyang@cqnu.edu.cn

Wang Chen
chenwangff@163.com

Yong Zhao
zhaoyongty@126.com

1 College of Mathematics, Sichuan University, Sichuan 610065, China
2 National Center for Applied Mathematics of Chongqing, Chongqing Normal University, Chongqing 401331, China
3 School of Mathematical Sciences, Chongqing Normal University, Chongqing 401331, China
4 College of Mathematics and Statistics, Chongqing Jiaotong University, Chongqing 610101, China
Keywords  Vector optimization · Conditional gradient method · Stationary point · Convergence · C-convexity

1 Introduction

In vector optimization, one considers a vector-valued function \( F : \mathbb{R}^n \to \mathbb{R}^m \), and the partial order defined by a closed, convex and pointed cone \( C \subseteq \mathbb{R}^m \) with nonempty interior, denoted by \( \text{int}(C) \). The partial order \( \preceq_C \) (\( \prec_C \)) induced by \( C \) (\( \text{int}(C) \)) in \( \mathbb{R}^m \) is given by \( y_1 \preceq_C y_2 \) (\( y_1 \prec_C y_2 \)) if and only if \( y_2 - y_1 \in C \) (\( y_2 - y_1 \in \text{int}(C) \)). We are interested in the following vector optimization problem:

\[
\begin{align*}
\min_C & \quad F(x) = (f_1(x), f_2(x), \ldots, f_m(x))^T \\
\text{s.t.} & \quad x \in \Omega,
\end{align*}
\]

where \( \Omega \subseteq \mathbb{R}^n \) is a nonempty set. Note that “\( \min_C \)” represents the optimum with respect to the cone \( C \). A point \( x \in \Omega \) is called a weakly efficient solution of problem (1) if there exists no \( x^* \in \Omega \) such that \( F(x^*) \prec_C F(x) \) (see [1]).

We will refer to (1) as an unconstrained vector optimization problem if \( \Omega = \mathbb{R}^n \). Otherwise, it is a constrained vector optimization problem. When \( m = 1 \) and \( \preceq_C \) is the usual linear order in \( \mathbb{R} \), the problem (1) is called the scalar optimization problem. When \( C = \mathbb{R}^m_+ \), where \( \mathbb{R}^m_+ \) is the non-negative orthant of \( \mathbb{R}^m \), the problem (1) is called the multicriteria or multiobjective optimization problem. Some practical applications of multiobjective optimization can be found in engineering [2], finance [3], environment analysis [4], management science [5], machine learning [6], etc. In recent years, many iterative methods for solving scalar optimization problems have been extended to the multiobjective optimization case, such as steepest descent [7], Newton [8, 9], subgradient [10], quasi-Newton [11], trust region [12] methods, and so on. It is worth mentioning that the vast majority of real-world problems formulated as vector-valued problems cope with the point-wise partial order \( \mathbb{R}^m_+ \). However, as reported in [13–15], there are many others that require preference orders induced by closed convex cones rather than the non-negative orthant. Such cones have been recently analyzed in real-life problems, for instance, the issues concerning portfolio selection in security markets [16, 17], the vector approximation problem and the cooperative \( n \)-player differential games [1]. Therefore, it is necessary to focus our attention on vector optimization problems.

One of the main solution strategies for vector optimization problems is the so-called scalarization method whose core idea is to convert a target vector optimization problem into a scalar one (see [1, 18, 19]), in such a way that the optimal solution of the new problem is also a solution for the original one. Another strategy is the iterative methods, which directly extend the existing scalar and multiobjective optimization methods to the vector optimization case. For example, the proximal point method used in scalar optimization has been successfully extended by Villacorta et al. in [20] to solve vector optimization problems on a non-polyhedral set. Chen [21] proposed a class of generalized viscosity methods to solve a vector optimization problem with a nonsmooth objective function. Bello Cruz [22] considered an extension of the
projected subgradient method to solve convex-constrained vector optimization problems. In [23, 24], the authors proposed some nonlinear conjugate gradient methods for solving unconstrained vector optimization problems. The vector extensions of the Fletcher–Reeves [25], conjugate descent [26], Dai–Yuan [27], Polak–Ribiére–Polyak [28, 29], Hestenes–Stiefel [30] and Hager–Zhang [31] parameters were considered. Graña Drummond et al. [14, 15] provided an extension of the projected gradient method to solve constrained vector optimization problems. In addition, based on Fliege and Svaiter’s work [7], Graña Drummond and Svaiter [13] regained the steepest descent method for solving unconstrained vector optimization problems. Compared with [7], the remarkable feature in [13] is that the subproblem is given with the help of a gauge function, which has a more general form. Note that when $C = \mathbb{R}^m_+$, vector versions of the projected gradient and steepest descent methods are consistent with the ones in [7] and [32], respectively. Following Graña Drummond and Svaiter’s [13] research path, vector versions of the Newton method [33, 34] have been proposed for solving vector optimization problems. It is worth noting that, as far as we know, Chen et al. [35], Chuong and Yao [36], Bonnel et al. [37], Chuong [38], and Boţ and Grad [39] also proposed some extension methods for solving vector optimization problems in infinite-dimensional settings. Based on the above analysis, the use of extensions of the iterative methods in scalar/multiobjective optimization to vector optimization is currently promising research.

We should be noted that most of the aforementioned methods for multiobjective/vector optimization problems are equipped with the monotone line search rule. It is well-known that, in scalar optimization, the enforcing monotonicity of function values makes the corresponding methods converge slower in the minimization process, especially when the iterate is in the bottom of a narrow curved valley, as reported in [40]. To overcome the shortcoming caused by the monotone search rule, numerous nonmonotone line methods were proposed and then verified numerically in the literature (see [40–43], but not limited to them). In recent years, studies on algorithms based on nonmonotone schemes for solving multiobjective optimization problems have attracted more attention. Mita et al. [44] extended the max-type [40] and average-type [41] nonmonotone line searches for scalar optimization to the multiobjective setting and then successfully applied them to multiobjective versions of the steepest descent and Newton methods. In [45], the authors considered a quasi-Newton method based on the nonmonotone line search technique of [41] for an unconstrained multiobjective optimization problem with the strongly convex objective function. Fazzio and Schuverdt [46] and Carrizzo [47] respectively applied the average-type and max-type nonmonotone line searches to the projected gradient method for constrained multiobjective optimization problems. Ramirez and Sottosanto [48] presented a trust region algorithm with the max-type nonmonotone technique for solving an unconstrained multiobjective optimization problem. Numerical experiments in [44, 45, 47, 48] indicate that the nonmonotone line searches tend to be more efficient than monotone ones. So far, to the best of our knowledge, there are few researches about extending the nonmonotone strategy to vector optimization where the partial order is induced by a general cone in $\mathbb{R}^m$. Therefore, it is necessary for us to study the nonmonotone techniques in vector optimization.
Very recently, Assunção et al. [49] introduced a multiobjective version of the classical conditional gradient (also known as Frank-Wolfe) method for constrained multiobjective optimization problems. Convergence analysis and iteration-complexity bounds of the method were detailedly studied under suitable assumptions. Their numerical experiments illustrate the effectiveness of the method. It is noteworthy that, in the conclusion part of [49], Assunção et al. put forward an interesting question:

“A natural question is whether it is possible to analyze the conditional gradient method for vector optimization problems, i.e., when the partial order in $\mathbb{R}^m$ is induced by other underlying cones instead of the non-negative orthant.”

The purpose of this paper is to take a step further on the direction of [49] and give a pertinent solution to the above question. Based on the research lines of literature [13–15, 33, 34, 36, 38], to solve problem (1), we directly and successfully extend the conditional gradient method proposed in [49]. At each iteration of our method, the descent direction is the difference between the current iterate and an optimal solution of an auxiliary subproblem. We should mention that the subproblem in our method is based on the well-known oriented distance function ([50, 51]). When the partial order cone is the non-negative orthant and the norm is the infinite norm, our subproblem’s form turns out to be the very same proposed in [49] for the multiobjective case. Our method considers three different step size strategies. Firstly, an Armijo line search rule is conducted to choose the step size at every iteration and ensures that the sequence of objective function values decreases in the partial order induced by the cone $C$. The convergence property that requires no convexity conditions on the objective function is established. Secondly, we analyze the proposed method with an adaptative step size rule. It is to emphasize that, in convergence analysis, we circumvent the intricate question of Lipschitz continuity of Jacobian for objective function by using an elegant cone convexity condition. The useful condition is inspired by the work of Bauschke et al. [52], and as far as we know, there is no research about extending the condition to vector optimization. Finally, based on the nonmonotone line search given by Gu and Mo [42] in the scalar context, we introduce a new nonmonotone technique in vector optimization and then prove that the line search also guarantees the convergence of our method. When the objective function is $C$-convex, we show that any sequence produced by this method converges to a weakly efficient solution. Finally, numerical experiments on convex and nonconvex multiobjective test problems are performed to illustrate the practical behavior of the method. When the order cone is not the non-negative orthant, we present the computational experiments to show how the method works on two vector optimization problems.

The outline of this paper is as follows. Section 2 presents some basic definitions, notations and auxiliary results which will be used in the sequel. Section 3 gives the general scheme for the conditional gradient method for constrained vector optimization problems. In Sect. 4, we analyze the convergence of the proposed method with three different step size strategies. In Sect. 5, some discussions about the subproblem and the adaptative step size are presented. Section 6 contains numerical experiments illustrating the performance of the proposed method. Finally, in Sect. 7, some conclusions and remarks are given.
2 Preliminaries

For a nonempty set $X \subseteq \mathbb{R}^m$, the boundary of $X$ is denoted by $\text{bd}(X)$. Let $I = \{1, 2, \ldots, m\}$. For a compact set $X$, the diameter of $X$ is given by $\text{diam}(X) = \max_{x, y \in X} \|x - y\|$, where $\| \cdot \|$ denotes the norm in $\mathbb{R}^m$.

We now recall the concept of oriented distance function (also called assigned distance function or Hiriart-Urruty function), which was proposed by Hiriart-Urruty [50] to investigate optimality conditions of nonsmooth optimization problems from the geometric point of view. The oriented distance function has been extensively used in several works, such as scalarization for vector optimization [51, 53], robustness for multiobjective optimization [54], optimality conditions for vector optimization [55] and for set-valued optimization [56], properties on set optimization [57, 58], and so on. Herein, we consider the oriented distance function in $\mathbb{R}^m$.

**Definition 1** [50, 51] Let $A$ be a subset of $\mathbb{R}^m$. The function $\Delta_A : \mathbb{R}^m \to \mathbb{R} \cup \{\pm \infty\}$, defined by

$$\Delta_A(y) = d_A(y) - d_{\mathbb{R}^m \setminus A}(y), \quad \forall y \in \mathbb{R}^m,$$

is called the oriented distance function, where $d_A(y) = \inf\{\|y - a\| : a \in A\}$ stands for the distance function from $y \in \mathbb{R}^m$ to the set $A$.

We give the following example to illustrate the function $\Delta_A$.

**Example 1** (i) If we consider the norm $\|y\|_2 = (\sum_{i=1}^{m} y_i^2)^{1/2}$ in $\mathbb{R}^m$ and $A = \{y \in \mathbb{R}^m : \|y\|_2 \leq 1\}$, then $\Delta_A(y) = \|y\|_2 - 1$.

(ii) Consider the norm $\|y\|_p = (\sum_{i=1}^{m} |y_i|^p)^{1/p}$ with $1 \leq p < \infty$ and $A = -\mathbb{R}_+^m$.

Then it holds that $d_A(y) = \|y^+\|_p$, where $y_i^+ = \max\{y_i, 0\}, i \in I$, and

$$d_{\mathbb{R}^m \setminus A}(y) = \begin{cases} 0, & \text{if } y_i \geq 0 \text{ for some } i, \\ -\max_{i \in I} y_i, & \text{if } y_i < 0. \end{cases}$$

Therefore,

$$\Delta_A(y) = \begin{cases} \|y^+\|_p, & \text{if } y \notin A, \\ \max_{i \in I} y_i, & \text{if } y \in A. \end{cases}$$

(iii) If we take the norm $\|y\|_\infty = \max_{i \in I} |y_i|$ and $A = -\mathbb{R}_+^m$, then $\Delta_A(y) = \max_{i \in I} y_i$.

(iv) Consider the norm $\|y\|_2$ in $\mathbb{R}^2$ and the partial order $C = \{y = (y_1, y_2) \in \mathbb{R}^2 : 0 \leq y_1 + y_2, 0 \leq y_2\}$. Let $A = -C$ and

$$B_1 = \{y = (y_1, y_2) \in \mathbb{R}^2 : y_1 \leq 0, y_2 > 0\},$$
$$B_2 = \{y = (y_1, y_2) \in \mathbb{R}^2 : y_1 - y_2 \leq 0, y_1 > 0\},$$
$$B_3 = \{y = (y_1, y_2) \in \mathbb{R}^2 : y_1 - y_2 > 0, y_1 + y_2 > 0\},$$
$$B_4 = \{y = (y_1, y_2) \in \mathbb{R}^2 : y_1 - y_2 > 0, y_1 + y_2 \leq 0\},$$
$$B_5 = \{y = (y_1, y_2) \in \mathbb{R}^2 : y_1 - y_2 \leq 0, y_2 \leq 0\}.
Clearly, \( A = B_4 \cup B_5 \) and \( \mathbb{R}^2 \setminus A = B_1 \cup B_2 \cup B_3 \). By a direct calculation, we have

\[
d_A(y) = \begin{cases} 
  y_2, & \text{if } y \in B_1, \\
  \|y\|_2, & \text{if } y \in B_2, \\
  |y_1 + y_2|/\sqrt{2}, & \text{if } y \in B_3
\end{cases}
\]

and

\[
d_{\mathbb{R}^2 \setminus A}(y) = \begin{cases} 
  |y_1 + y_2|/\sqrt{2}, & \text{if } y \in B_4, \\
  |y_2|, & \text{if } y \in B_5.
\end{cases}
\]

As pointed out by Ansari et al. [54, Remark 2.1], the oriented distance function can easily be implemented by using the fast marching method, fast sweeping method and level set methods.

In this paper, for our purposes, let \( A = -C \) in Definition 1. For the sake of convenience, in what follows, we will use

\[
\varphi_c(y) = \Delta - C(y), \quad \forall y \in \mathbb{R}^m.
\]  

According to [51, Proposition 3.2] and the fact that \( C \) is a closed, convex and pointed cone with nonempty interior, we immediately have the following properties related to \( \varphi_c \), which will be used in our subsequent analysis.

**Lemma 1** [51] Let \( \varphi_c(\cdot) \) be defined in (2). Then, the following statements hold:

(i) \( \varphi_c \) is a 1-Lipschitzian function;

(ii) \( \varphi_c(y) < 0 \) for any \( y \in -\text{int}(C) \), \( \varphi_c(y) = 0 \) for any \( y \in \text{bd}(-C) \), and \( \varphi_c(y) > 0 \) for any \( y \in \text{int}(\mathbb{R}^m \setminus (-C)) \);

(iii) \( \varphi_c \) is convex;

(iv) \( \varphi_c \) is positively homogeneous;

(v) For all \( y_1, y_2 \in \mathbb{R}^m \), \( \varphi_c(y_1 + y_2) \leq \varphi_c(y_1) + \varphi_c(y_2) \) and \( \varphi_c(y_1) - \varphi_c(y_2) \leq \varphi_c(y_1 - y_2) \);

(vi) Let \( y_1, y_2 \in \mathbb{R}^m \). If \( y_1 \preceq_C (\prec_C) y_2 \), then \( \varphi_c(y_1) \leq (\prec) \varphi_c(y_2) \).

From now on, we assume that \( F \) is a continuously differentiable function and the constraint set \( \Omega \subseteq \mathbb{R}^n \) is nonempty, compact and convex. Given \( x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n \), the Jacobian of \( F \) at \( x \), denoted by \( JF(x) \), is a matrix of order \( m \times n \) whose entries are defined by

\[
(JF(x))_{i,j} = \frac{\partial f_i}{\partial x_j}(x),
\]

where \( i \in I \) and \( j \in \{1, 2, \ldots, n\} \). We may represent it by

\[
JF(x) = [\nabla f_1(x) \ \nabla f_2(x) \ \ldots \ \nabla f_m(x)]^T, \quad x \in \mathbb{R}^n.
\]
A necessary, but not sufficient, first-order optimality condition for problem (1) at \( x \in \Omega \), is
\[
JF(x)(\Omega - x) \cap (-\text{int}(C)) = \emptyset, \tag{3}
\]
where \( JF(x)(\Omega - x) = \{JF(x)(s - x) : s \in \Omega\} \) and
\[
JF(x)(s - x) = ((\nabla f_1(x), s - x), (\nabla f_2(x), s - x), \ldots, (\nabla f_m(x), s - x))^\top.
\]
Obviously, (3) is equivalent to \( JF(x)(s - x) \not\in -\text{int}(C) \) for any \( s \in \Omega \).

**Definition 2** A point \( \hat{x} \in \Omega \) satisfying (3) is called a stationary point of problem (1).

**Remark 1** From Lemma 1(ii), we can obtain an equivalent characterization of a stationary point \( \hat{x} \) of problem (1), i.e.,
\[
\varphi_C(JF(\hat{x})(s - \hat{x})) \geq 0, \quad \forall s \in \Omega.
\]

**Remark 2** (i) If \( m = 1 \) and \( C = \mathbb{R}_+^1 \), then we retrieve the classical stationary condition for constrained scalar optimization problem, i.e., \( (\nabla f_1(\hat{x}), s - \hat{x}) \geq 0 \) for all \( s \in \Omega \).
(ii) If we consider the norm \( \| \cdot \|_\infty \) and \( C = \mathbb{R}_+^m \), then by Remark 1 and Example 1(iii), Definition 2 is the same as the notion presented in [49, pp.744].

**Remark 3** Note that if \( \hat{x} \in \Omega \) is not a stationary point of problem (1), then there exists \( \hat{s} \in \Omega \) such that \( JF(\hat{x})(\hat{s} - \hat{x}) \in -\text{int}(C) \), i.e., \( \varphi_C(JF(\hat{x})(\hat{s} - \hat{x})) < 0 \) from Lemma 1(ii). In this case, as analyzed in [33, pp.665], we can assert that \( \hat{s} - \hat{x} \) is a descent direction for \( F \).

We now recall the concept of \( C \)-convexity of \( F \), which is presented in [1, Definition 2.4]. The objective function \( F \) is called \( C \)-convex on \( \Omega \) if
\[
F(\lambda x + (1 - \lambda)y) \preceq_C \lambda F(x) + (1 - \lambda)F(y),
\]
for all \( x, y \in \Omega \) and all \( \lambda \in [0, 1] \). Since \( F \) is continuously differentiable, an equivalent characterization of \( C \)-convexity of \( F \) is
\[
JF(y)(x - y) \preceq_C F(x) - F(y)
\]
for all \( x, y \in \Omega \) (see [1, Theorem 2.20]).

We conclude this section by giving the relationship between stationary point and weakly efficient solution. The proof of this property can be analogously analyzed from [13, pp.410], and we omit the process here.

**Theorem 1** (i) If \( \hat{x} \in \Omega \) is a weakly efficient solution of problem (1), then \( \hat{x} \in \Omega \) is a stationary point.
(ii) If \( F \) is \( C \)-convex on \( \Omega \) and \( \hat{x} \in \Omega \) is a stationary point of problem (1), then \( \hat{x} \) is a weakly efficient solution.
3 Conditional gradient method for vector optimization

In this section, we propose a general scheme of conditional gradient (for short, CondG) method for the vector optimization problem (1).

For a given \( x \in \Omega \), we introduce a useful auxiliary function \( \psi_x : \Omega \rightarrow \mathbb{R} \) defined by

\[
\psi_x(s) = \varphi_C(JF(x)(s-x)), \quad s \in \Omega.
\] (4)

For \( x \in \Omega \), in order to obtain the descent direction for \( F \) at \( x \), we need to consider the following auxiliary scalar optimization problem:

\[
\min_{s \in \Omega} \psi_x(s).
\] (5)

It follows from Lemma 1(iii) that \( \psi_x \) defined in (4) is a convex function. This, combined with the fact that \( \Omega \) is a nonempty, compact and convex set, gives that problem (5) admits an optimal solution (possibly not unique) on \( \Omega \). We denote the optimal solution of problem (5) by \( s(x) \), i.e.,

\[
s(x) \in \arg\min_{s \in \Omega} \psi_x(s),
\] (6)

and the optimal value of problem (5) is denoted by \( v(x) \), i.e.,

\[
v(x) = \psi_x(s(x)).
\] (7)

According to Remark 3, we formally give the search direction for the objective function \( F \) at \( x \).

**Definition 3** For any given point \( x \in \Omega \), the search direction of the conditional gradient method for \( F \) at \( x \) is defined as

\[
d(x) = s(x) - x,
\]
where \( s(x) \) is given by (6).

The following property gives a characterization of stationarity in terms of \( v(\cdot) \), which is crucial for convergence analysis and the stopping criteria of our algorithm.

**Proposition 2** Let \( v : \Omega \rightarrow \mathbb{R} \) be defined in (7). Then, the following statements hold:

(i) \( v(x) \leq 0 \) for all \( x \in \Omega \);

(ii) \( x \in \Omega \) is a stationary point of problem (1) if and only if \( v(x) = 0 \).

**Proof** (i) Since \( x \in \Omega \), by (6) and (7) and observing that \( \varphi_C(0) = 0 \), we have

\[
v(x) = \min_{s \in \Omega} \psi_x(s) \leq \psi_x(x) = \varphi_C(JF(x)(x-x)) = \varphi_C(0).
\]
(ii) Necessity. Suppose that \( x \in \Omega \) is a stationary point of problem (1). Then, it follows from Remark 1 that \( \varphi_c(JF(x)(s - x)) \geq 0 \) for any \( s \in \Omega \). By (6), we have \( s(x) \in \Omega \). Hence, \( v(x) = \varphi_c(JF(x)(s(x) - x)) \geq 0 \). This, combined with (i), yields that \( v(x) = 0 \).

Sufficiency. Let \( v(x) = 0 \). According to (7), we obtain

\[
0 = v(x) \leq \psi_x(s) = \varphi_c(JF(x)(s - x))
\]

for all \( s \in \Omega \), which implies that \( x \) is a stationary point of problem (1).

\[\square\]

Remark 4 It is obvious from Proposition 2 that \( x \) is not a stationary point of problem (1) if and only if \( v(x) < 0 \).

Proposition 3 Let \( v : \Omega \rightarrow \mathbb{R} \) be defined in (7). Then, \( v \) is continuous on \( \Omega \).

Proof Take \( x \in \Omega \) and let \( \{x^k\} \) be a sequence in \( \Omega \) such that \( \lim_{k \rightarrow \infty} x^k = x \). In order to obtain the continuity of \( v \) on \( \Omega \), it is sufficient to prove that \( \lim_{k \rightarrow \infty} v(x^k) = v(x) \), i.e.,

\[
\limsup_{k \rightarrow \infty} v(x^k) \leq v(x) \leq \liminf_{k \rightarrow \infty} v(x^k). \quad (8)
\]

Since \( s(x) \in \Omega \), using (6) and (7), we can obtain for all \( k \),

\[
v(x^k) = \varphi_c(JF(x^k)(s(x^k) - x^k)) \leq \varphi_c(JF(x^k)(s(x) - x^k)). \quad (9)
\]

Since \( F \) is continuously differentiable and \( \varphi_c \) is continuous as presented in Lemma 1(i), taking \( \limsup_{k \rightarrow \infty} \) on both sides of inequality in (9), we have

\[
\limsup_{k \rightarrow \infty} v(x^k) \leq \varphi_c(JF(x)(s(x) - x)) = v(x).
\]

Let us show that \( v(x) \leq \liminf_{k \rightarrow \infty} v(x^k) \). Obviously, we have

\[
v(x) = \min \{\psi_x(s), s \in \Omega\} \\
\leq \psi_x(s(x^k)) \\
= \varphi_c(JF(x)(s(x^k) - x)) \\
= \varphi_c(JF(x)(s(x^k) - x^k + x^k - x)) \\
= \varphi_c(JF(x)(s(x^k) - x^k) + JF(x)(x^k - x)) \\
\leq \varphi_c(JF(x)(s(x^k) - x^k)) + \varphi_c(JF(x)(x^k - x)),
\]

(10)
where the last inequality follows from Lemma 1(v). Taking \( \liminf_{k \to \infty} \) in (10), we get

\[
v(x) \leq \liminf_{k \to \infty} \varphi_c(JF(x)(s(x^k) - x^k))
= \liminf_{k \to \infty}(v(x^k) + \varphi_c(JF(x)(s(x^k) - x^k)) - \varphi_c(JF(x^k)(s(x^k) - x^k)))
\leq \liminf_{k \to \infty}(v(x^k) + \|JF(x)(s(x^k) - x^k) - JF(x^k)(s(x^k) - x^k)\|)
= \liminf_{k \to \infty}(v(x^k) + \|(JF(x) - JF(x^k))(s(x^k) - x^k)\|)
\leq \liminf_{k \to \infty}(v(x^k) + \|JF(x) - JF(x^k)\|\|s(x^k) - x^k\|),
\]

where the second inequality follows from Lemma 1(i). Since \( s(x^k), x^k \in \Omega \), it follows that \( \|s(x^k) - x^k\| \leq \text{diam}(\Omega) < \infty \). This, combined with the continuously differentiability of \( F \) and (11), we get \( v(x) \leq \liminf_{k \to \infty} v(x^k) \). Altogether, (8) holds. Consequently, \( v \) is continuous on \( \Omega \). \( \square \)

Based on the previous discussions, we are now ready to describe the general scheme of the conditional gradient algorithm for solving problem (1).

**CondG algorithm.**

**Step 0** Choose \( x^0 \in \Omega \). Compute \( s(x^0) \) and \( v(x^0) \) and initialize \( k \leftarrow 0 \).

**Step 1** If \( v(x^k) = 0 \), then **stop**.

**Step 2** Compute \( d(x^k) = s(x^k) - x^k \).

**Step 3** Compute a step size \( t_k \in (0, 1] \) by a step size strategy and set \( x^{k+1} = x^k + t_k d(x^k) \).

**Step 4** Compute \( s(x^{k+1}) \) and \( v(x^{k+1}) \), set \( k \leftarrow k + 1 \), and go to **Step 1**.

At \( k \)-th iteration, we solve problem (5) with \( x = x^k \). Let us call \( s(x^k) \) and \( v(x^k) \) the optimal solution and optimal value of problem (5) at \( k \)-th iteration, respectively. The descent direction at \( k \)-th iteration is computed by \( d(x^k) = s(x^k) - x^k \). If \( v(x^k) \neq 0 \), then we can use \( d(x^k) \) with a step size strategy to look for a new solution \( x^{k+1} \) which improves \( x^k \). For convenience, we can also use \( s^k, v^k \) and \( d^k \) instead of \( s(x^k), v(x^k) \) and \( d(x^k) \), receptively.

Observe that the CondG algorithm can terminate with a stationary point in a finite number of iterations or produce an infinite sequence of nonstationary points. In convergence analysis, we will assume that \( \{x^k\}, \{s^k\}, \{d^k\} \) and \( \{t^k\} \) are infinite sequences. Therefore, using Remark 4, we have \( v^k < 0 \) for all \( k \). An elementary fact that the proposed algorithm generates feasible sequences is given below.

**Remark 5** It is noteworthy that the sequence \( \{x^k\} \) generated by the CondG algorithm contains in \( \Omega \). Indeed, since \( x^0 \in \Omega, s^k \in \Omega \) and \( t_k \in (0, 1] \), it follows from inductive arguments that \( x^k \in \Omega \) for all \( k \).
4 Convergence analysis

It is worth noting that the choice for computing the step size \( t_k \) at Step 3 is crucial. In this section, we establish the main convergence properties of the CondG algorithm with three different step size strategies.

4.1 Analysis of CondG algorithm with Armijo step size

Armijo step size. Let \( \beta \in (0, 1) \) and \( \delta \in (0, 1) \). Choose \( t_k \) as the largest value in \( \{ \delta^0, \delta^1, \delta^2, \ldots \} \) such that

\[
F(x^k + t_k d^k) \preceq C F(x^k) + \beta t_k JF(x^k) d^k. \tag{12}
\]

Other vector optimization methods, including steepest descent, Newton and projected gradient methods, that use the Armijo rule can be found in [13–15, 34, 36, 38]. The following proposition allows us to implement the line search (12) at Step 3 of the CondG algorithm.

**Proposition 4** Let \( \beta \in (0, 1) \) and \( s(x) \) be defined in (6) and \( JF(x)(s(x) - x) \prec_C 0 \). Then, there exists some \( \hat{t} \in (0, 1] \) such that

\[
F(x + t(s(x) - x)) \prec_C F(x) + t\beta JF(x)(s(x) - x),
\]

for any \( t \in (0, \hat{t}] \).

**Proof** It is analogous to the proof of [13, Proposition 2.1]. \( \square \)

We present some properties related to the points which are iterated by the CondG algorithm with the Armijo step size condition (12).

**Proposition 5** For all \( k \), we have

(i) \( F(x^{k+1}) \prec_C F(x^k) \).

(ii) \( \sum_{i=0}^k t_i |v^i| \leq \beta^{-1}(\varphi_C(F(x^0)) - \varphi_C(F(x^{k+1}))) \).

**Proof** (i) follows from (12) and the nonstationarity of \( x^k \).

(ii) For any \( i \), by (12) and Lemma 1(iv)–(vi), we have

\[
\begin{align*}
\varphi_C(F(x^{i+1})) &\leq \varphi_C(F(x^i)) + t_i \beta JF(x^i)(s^i - x^i) \\
&\leq \varphi_C(F(x^i)) + \varphi_C(t_i \beta JF(x^i)(s^i - x^i)) \\
&= \varphi_C(F(x^i)) + t_i \beta \varphi_C(JF(x^i)(s^i - x^i)) \\
&= \varphi_C(F(x^i)) + t_i \beta v^i. \tag{13}
\end{align*}
\]

From (13) and observing that \( v^i < 0 \), we obtain

\[
t_i |v^i| = -t_i v^i \leq \frac{\varphi_C(F(x^i)) - \varphi_C(F(x^{i+1}))}{\beta}.
\]
Summing the above inequality from \( i = 0 \) to \( i = k \), the result is immediately derived. \( \square \)

**Theorem 6** Let \( \{x^k\} \) be a sequence produced by the CondG algorithm with the Armijo step size condition (12). Then, every accumulation point of \( \{x^k\} \) is a stationary point of problem (1).

**Proof** Let \( \hat{x} \in \mathbb{R}^n \) be an accumulation point of the sequence \( \{x^k\} \). Clearly, \( \hat{x} \in \Omega \) follows from Remark 5 and the closedness of \( \Omega \). Then, there exists a subsequence \( \{x^{k_j}\}_j \) of \( \{x^k\} \) such that

\[
\lim_{j \to \infty} x^{k_j} = \hat{x}. \tag{14}
\]

From Proposition 3 and (14), we have \( v(x^{k_j}) \to v(\hat{x}) \) whenever \( j \to \infty \). Here, it is sufficient to show that \( v(\hat{x}) = 0 \) in view of Proposition 2(ii).

Let \( k = k_j \) in Proposition 5(ii). Then

\[
\sum_{i=0}^{k_j} t_i |v(x^i)| \leq \frac{\varphi_c(F(x^0)) - \varphi_c(F(x^{k_j+1}))}{\beta}.
\]

Taking \( \lim_{j \to \infty} \) on both sides of the above inequality, we get \( \sum_{i=0}^{\infty} t_i |v(x^i)| < \infty \), which implies that \( \lim_{k \to \infty} t_k v(x^k) = 0 \), and in particular,

\[
\lim_{j \to \infty} t_{k_j} v(x^{k_j}) = 0. \tag{15}
\]

Since \( t_k \in (0, 1] \) for all \( k \), we have the following two alternatives:

\[
(\text{a}) \quad \limsup_{j \to \infty} t_{k_j} > 0 \quad \text{or} \quad (\text{b}) \quad \limsup_{j \to \infty} t_{k_j} = 0. \tag{16}
\]

We first suppose that (16)(a) holds. Then, there exists a subsequence \( \{t_{k_{j_i}}\}_i \) converging to some \( \hat{t} > 0 \). And from (14), we have \( \lim_{i \to \infty} x^{k_{j_i}} = \hat{x} \). Thus, (15) implies that \( \lim_{i \to \infty} t_{k_{j_i}} v(x^{k_{j_i}}) = 0 \), and furthermore, \( \lim_{i \to \infty} v(x^{k_{j_i}}) = 0 \). This, combined with Proposition 3, gives that \( 0 = \lim_{i \to \infty} v(x^{k_{j_i}}) = v(\hat{x}) \).

Consider (16)(b). Feasibility of \( \hat{x} \) and Proposition 2 gives

\[
v(\hat{x}) \leq 0. \tag{17}\]

Clearly, \( x^{k_j}, s(x^{k_j}) \in \Omega \) and

\[
\|d(x^{k_j})\| = \|s(x^{k_j}) - x^{k_j}\| \leq \text{diam}(\Omega) < \infty
\]

for all \( j \), which implies that the sequence \( \{d(x^{k_j})\}_j \) is bounded. Therefore, we can take subsequences \( \{x^{k_{j_i}}\}_i, \{d(x^{k_{j_i}})\}_i \) and \( \{t_{k_{j_i}}\}_i \) converging to \( \hat{x}, d(\hat{x}) \) and 0, respectively.

\( \square \)
Now, take some fixed but arbitrary $l \in \mathbb{N}$, where $\mathbb{N}$ denotes the set of natural numbers. From $\lim_{i \to \infty} t_{k_{ji}} = 0$, we have $t_{k_{ji}} < \delta^l$ for $i$ large enough. This shows that the Armijo condition (12) is not satisfied at $x^{k_{ji}}$ for $t = \delta^l$, that is,

$$F(x^{k_{ji}} + \delta^l d(x^{k_{ji}})) \not\in C (x^{k_{ji}} + \delta^l \beta J F(x^{k_{ji}})d(x^{k_{ji}})),$$

or, equivalently,

$$F(x^{k_{ji}} + \delta^l d(x^{k_{ji}})) - F(x^{k_{ji}}) - \delta^l \beta J F(x^{k_{ji}})d(x^{k_{ji}}) \not\in -C,$$

which means that

$$F(x^{k_{ji}} + \delta^l d(x^{k_{ji}})) - F(x^{k_{ji}}) - \delta^l \beta J F(x^{k_{ji}})d(x^{k_{ji}}) \in \mathbb{R}^m \setminus (-C)$$

where the equality holds in view of the closedness of $C$. By (18) and Lemma 1(ii), we have

$$\varphi_C (F(x^{k_{ji}} + \delta^l d(x^{k_{ji}})) - F(x^{k_{ji}}) - \delta^l \beta J F(x^{k_{ji}})d(x^{k_{ji}})) > 0. \quad (19)$$

Since $F$ is continuously differentiable and $\varphi_C$ is continuous, taking $\lim_{i \to \infty}$ in (19), we obtain

$$\varphi_C (F(\hat{x} + \delta^l d(\hat{x})) - F(\hat{x}) - \delta^l \beta J F(\hat{x})d(\hat{x})) \geq 0, \quad \forall l \in \mathbb{N}. \quad (20)$$

According to (20), Proposition 4 and Lemma 1(ii), we can obtain $J F(\hat{x})d(\hat{x}) \not\in C 0$, i.e., $J F(\hat{x})d(\hat{x}) \in \mathbb{R}^m \setminus (-\text{int}(C))$. Thus, we have $v(\hat{x}) = \varphi_C (J F(\hat{x})d(\hat{x})) \geq 0$ from Lemma 1(ii). This, combined with (17), yields that $v(\hat{x}) = 0$. 

It follows from Theorems 1 and 6 that the following result holds.

**Theorem 7** If $F$ is $C$-convex on $\Omega$, then the sequence $\{x^k\}$ produced by the CondG algorithm with the Armijo step size condition (12) converges to a weakly efficient solution of problem (1).

### 4.2 Analysis of CondG algorithm with adaptative step size

In this section, we analyze the convergence of the CondG algorithm with an adaptative step size rule. The difference from the work [49] is that we not only consider the general partial order but also do not require the Jacobian $J F$ of the objective function to be Lipschitz continuous.

In this sequel, we let $e = (e_1, e_2, \ldots, e_m)^\top \in \text{int}(C)$. Let us recall the valuable concept that the Jacobian $J F$ of $F$ is Lipschitz continuous with Lipschitz constant $L_F > 0$ on $\Omega$ if

$$\|J F(x) - J F(y)\| \leq L_F \|x - y\| \quad (21)$$
for all \( x, y \in \Omega \). It is notable that the condition (21) was widely used in convergence analysis of algorithms for multiobjective/vector optimization, see, e.g., [23, 24, 49]. In subsequent convergence analysis, we shall use the following condition:

\[(A) \quad \exists \, L > 0 \quad \text{s.t.} \quad \frac{L}{2} \| \cdot \|_2^2 e - F(\cdot) \text{ is } C\text{-convex on } \Omega.\]

Actually, the introduction of this condition is inspired by the Lipschitz-like/convexity (LC) condition proposed by Bauschke et al. in [52].

**Remark 6** Let \( m = 1, C = \mathbb{R}_+ \) and \( e = 1 \). In such case, if we set \( h = \| \cdot \|_2^2 / 2 \) and \( g = f_1 \) in [52], then the condition (A) can be regarded as the LC condition. Although it was assumed that \( g \) is convex in the LC condition [52], the convexity assumption of \( g \) plays no role in the LC condition, and \( g \) can be nonconvex, as already observed in [59]. Therefore, our condition (A) does not require \( C\)-convexity of \( F \).

We will see that, in the following Lemma 2, the mere translation of condition (A) into its first-order characterization immediately gives the new and key vector version of the descent lemma. For convenience, we let

\[
G(\cdot) = (g_1(\cdot), g_2(\cdot), \ldots, g_m(\cdot)) = \frac{L}{2} \| \cdot \|_2^2 e - F(\cdot).
\]

**Lemma 2** The condition (A) is equivalent to

\[
F(y) - F(x) \preceq_C JF(x)(y - x) + \frac{L}{2} \| y - x \|_2^2 e, \quad \forall x, y \in \Omega. \tag{22}
\]

**Proof** For any \( x, y \in \Omega \), the function \( G(\cdot) \) is \( C\)-convex on \( \Omega \) if and only if

\[
JG(x)(y - x) \preceq_C G(y) - G(x). \tag{23}
\]

For \( JG(x)(y - x) \) in (23), by a simple calculation, we have

\[
JG(x)(y - x) = L \langle x, y \rangle e - L \| x \|_2^2 e - JF(x)(y - x). \tag{24}
\]

From the notion of \( G(\cdot) \), (23) and (24), we obtain

\[
F(y) - F(x) \preceq_C JF(x)(y - x) + \frac{L}{2} \| y \|_2^2 e - \frac{L}{2} \| x \|_2^2 e - L \langle x, y \rangle e + L \| x \|_2^2 e
\]

\[
= JF(x)(y - x) + \frac{L}{2} (\| y \|_2^2 + \| x \|_2^2 - 2 \langle x, y \rangle) e
\]

\[
= JF(x)(y - x) + \frac{L}{2} \| y - x \|_2^2 e,
\]

and thus the proof is complete. \( \square \)

From condition (A) and Lemma 2, we have the following some remarks.
Remark 7  
(i) If \( m = 1, C = \mathbb{R}_+ \) and \( e = 1 \), then Lemma 2 becomes [52, Lemma 1] by Remark 6.

(ii) Let \( C = \mathbb{R}_+^m \) and \( e = (1, 1, \ldots, 1)^\top \). In this case, as reported in [49, Lemma 1], by using the same idea in the proof [60, Lemma 2.4.2], we can also obtain (22) with \( L = L_F \) from the Lipschitz continuity of \( JF \). Therefore, the Lipschitz continuity of \( JF \) implies that the condition (A) in view of Lemma 2. However, the following example shows that the reverse relationship does not necessarily hold.

Example 2  Let \( \Omega = [0, 1] \) and

\[
F(x) = \left(1 - \frac{2}{3} x^{3/2}, \left(x - \frac{1}{2}\right)^2\right)^\top, \quad \forall x \in \Omega.
\]

The image of \( F \) is shown in Fig. 1a. Consider \( C = \mathbb{R}_+^2 \) and \( e = (1, 1)^\top \). It is easy to check that for \( L = 2 \), \( G \) is \( \mathbb{R}_+^2 \)-convex on \( \Omega \) (see Fig. 1b). However, \( JF \) is not Lipschitz continuous on \( \Omega \).

Remark 8  For a general partial order \( C \) rather than the non-negative orthant \( \mathbb{R}_+^m \), the relationships between Lipschitz continuity of \( JF \) and condition (A) seem less readily available. But we have the same observation as Remark 7 that the condition (A) does not imply the Lipschitz continuity of \( JF \), the following example shows this case. Therefore, we propose an open question in this paper: How to obtain the relations between them under some suitable assumptions for a general partial order \( C \)?

Example 3  We continue to consider Example 2. Let \( C = \{y \in \mathbb{R}^2 : 0 \leq y_1 + y_2, 0 \leq y_1\} \) and \( e = (1, 1)^\top \). It is easy to check that for \( L = 1 \), \( G \) is \( C \)-convex but fails to be \( \mathbb{R}_+^2 \)-convex on \( \Omega \) (see Figure 1c). However, \( JF \) is not Lipschitz continuous.

Remark 9  Lemma 2 implies that, for any \( x, y \in \Omega \),

\[
\varphi_C(F(y)) - \varphi_C(F(x)) \leq \psi_x(y) + \frac{L}{2} \|y - x\|^2 \varphi_C(e). \tag{25}
\]
Indeed, from (22) and Lemma 1(iv)–(vi), we have
\[
\varphi_c(F(y) - F(x)) \leq \varphi_c(J_F(x) - x) + \frac{L}{2} \|y - x\|_2^2 e \leq \varphi_c(J_F(x) - x) + \frac{L}{2} \|y - x\|_2^2 e \tag{26}
\]
\[
= \varphi_c(J_F(x)(y - x)) + \frac{L}{2} \|y - x\|_2^2 \varphi_c(e).
\]
Obviously, according to Lemma 1(vi), it holds that
\[
\varphi_c(F(y)) - \varphi_c(F(x)) \leq \varphi_c(F(y) - F(x)) \tag{27}
\]
Therefore, it immediately follows from (26), (27) and (4) that (25) holds.

We now give the adaptive step size strategy with the help of condition (A).

**Adaptive step size.** Assume that the condition (A) holds. Define the step size as
\[
t_k = \min \left\{ 1, -\frac{v_k}{L \|d_k\|_2^2} \right\} \tag{28}
\]
Since \(v(x) < 0\) and \(s(x) \neq x\) for nonstationary points, the adaptive step size for the CondG algorithm is well-defined. Before giving our convergence result, we first present an important property that is essential for showing convergence analysis.

**Lemma 3** Let \(e \in \text{int}(C)\) with \(\varphi_c(e) < 2\). Suppose that the condition (A) holds and \(\{x^k\}\) is a sequence produced by the CondG algorithm with the adaptive step size. Then, for all \(k\), it holds that
\[
\varphi_c(F(x^{k+1})) - \varphi_c(F(x^k)) \leq \frac{\varphi_c(e) - 2}{2} \min \left\{ \frac{(v_k)^2}{L(\text{diam}(\Omega))^2}, -v_k \right\}. \tag{29}
\]

**Proof** Let \(x^{k+1} = x^k + t_k d^k\), where \(d^k = s^k - x^k\) and
\[
t_k = \min \left\{ 1, -\frac{v_k}{L \|d_k\|_2^2} \right\}. \tag{30}
\]
Since the condition (A) holds, then by (25) invoked with \(x = x^k\) and \(y = x^{k+1}\), we have
\[
\varphi_c(F(x^{k+1})) - \varphi_c(F(x^k)) \leq \psi_{x^k}(x^k + t_k (s^k - x^k)) + \frac{L}{2} t_k^2 \|d^k\|_2^2 \varphi_c(e)
\]
\[
= t_k \psi_{x^k}(s^k) + \frac{L}{2} t_k^2 \|d^k\|_2^2 \varphi_c(e) \tag{31}
\]
\[
= t_k v_k^k + \frac{L}{2} t_k^2 \|d^k\|_2^2 \varphi_c(e).
\]
where the first equality holds in view of (4) and Lemma 1(iv). According to (30), there are two options:

**Case 1.** Let $t_k = 1$. This, combined with (30), gives

$$L \|d^k\|_2^2 \leq -v^k.$$  

(32)

By (31) and (32), we obtain

$$\varphi_C(F(x^{k+1})) - \varphi_C(F(x^k)) \leq \frac{2 - \varphi_C(e)}{2} v^k.$$  

(33)

**Case 2.** Let $t_k = -\frac{v^k}{(L \|d^k\|_2^2)}$. Clearly, $\|d^k\|_2 = \|x^k - x_k\|_2 \leq \text{diam}(\Omega)$. This, together with (31) and the fact that $\varphi_C(e) < 2$, yields

$$\varphi_C(F(x^{k+1})) - \varphi_C(F(x^k)) \leq \frac{\varphi_C(e) - 2}{2} \frac{(v^k)^2}{L \|d^k\|_2^2} \leq \frac{\varphi_C(e) - 2}{2} \frac{(v^k)^2}{L(\text{diam}(\Omega))^2}.$$  

(34)

Therefore, (29) is directly derived by (33) and (34).

\[\square\]

**Theorem 8** Suppose that all conditions of Lemma 3 hold. Then, every accumulation point of \(\{x_k\}\) is a stationary point of problem (1).

**Proof** From (29) and $\varphi_C(e) < 2$ and observing that $v^k < 0$, we have for all $k$,

$$\varphi_C(F(x^{k+1})) - \varphi_C(F(x^k)) \leq \frac{\varphi_C(e) - 2}{2} \min \left\{ \frac{(v^k)^2}{L(\text{diam}(\Omega))^2}, -v^k \right\} < 0,$$  

(35)

i.e., $\varphi_C(F(x^{k+1})) < \varphi_C(F(x^k))$, which implies that \(\{\varphi_C(F(x^k))\}\) is nonincreasing for all $k$. By Remark 5 and the continuity of $F$, there exits $\bar{F} \in \mathbb{R}^m$ such that $\bar{F} \preceq_C F(x^k)$ for all $k$, it follows from Lemma 1(vi) that $\varphi_C(\bar{F}) \leq \varphi_C(F(x^k))$ for all $k$. Therefore, we conclude that the sequence $\{\varphi_C(F(x^k))\}$ is convergent. This obviously means that

$$\lim_{k \to \infty} (\varphi_C(F(x^{k+1})) - \varphi_C(F(x^k))) = 0.$$  

(36)

Taking $\lim_{k \to \infty}$ in (35), and then combining with (36), we have

$$\lim_{k \to \infty} v(x^k) = 0.$$  

(37)

From Proposition 2(ii), Proposition 3 and (37), we obtain that each accumulation point of $\{x^k\}$ is a stationary point of problem (1).

\[\square\]

**Remark 10** If we consider the norm $\|\cdot\|_\infty$, $C = \mathbb{R}^m_+$ and $e = (1, 1, \ldots, 1) \top \in \text{int}(\mathbb{R}^m_+)$, then $\varphi_C(e) = 1 < 2$, and thus Theorem 8 is an improvement of the corresponding convergence result presented in [49].

\[\square\]
According to Theorems 1 and 8, we have the following result.

**Theorem 9** If \( F \) is \( C \)-convex on \( \Omega \) and all conditions of Theorem 8 hold, then the sequence \( \{x^k\} \) produced by the CondG algorithm with the adaptative step size strategy (28) converges to a weakly efficient solution of problem (1).

### 4.3 Analysis of CondG algorithm with nonmonotone line search

In this section, we extend a nonmonotone line search for scalar optimization proposed in [42] to the vector setting and establish the convergence property of the CondG algorithm with it.

In the sequel, we denote
\[
D^k = (D^k_1, D^k_2, \ldots, D^k_m)^T \in \mathbb{R}^m.
\]

**Nonmonotone line search.** Let \( \beta \in (0, 1), \delta \in (0, 1) \) and \( \eta_{\text{max}} \in [0, 1) \). Choose \( t_k \) as the largest value in \( \{\delta^0, \delta^1, \delta^2, \ldots\} \) such that
\[
F(x^k + t_k d^k) \preceq_C D^k + \beta t_k JF(x^k)d^k,
\]
where
\[
\left\{
\begin{array}{ll}
D^0 = F(x^0) \\
D^k = \eta_k D^{k-1} + (1 - \eta_k) F(x^k), & k \geq 1
\end{array}
\right.
\]
with \( 0 \leq \eta_k \leq \eta_{\text{max}} < 1 \).

**Remark 11** The current nonmonotone term in (38) is a convex combination of the previous nonmonotone term and the current objective function value, which is completely different from the maximum of recent objective function values and an average of the successive objective function values proposed by [44]. Note that \( \eta_k \) controls the degree of nonmonotonicity in \( D^k \). Moreover, (38) may be viewed as a generalization of (12) since the particular choice \( \eta_k = 0 \) recovers (12).

**Lemma 4** Let \( \{x^k\} \) be the sequence generated by the CondG algorithm with the nonmonotone line search (38). If \( JF(x^k)d^k \preceq_C 0 \), then \( F(x^k) \preceq_C D^k \) for all \( k \geq 0 \).

**Proof** We proceed by induction. Obviously, \( F(x^k) = D^k \) for \( k = 0 \). Assume that the conclusion holds for \( k > 0 \). We shall prove that \( F(x^{k+1}) \preceq_C D^{k+1} \). By \( JF(x^k)d^k \preceq_C 0 \) and (38), one has
\[
F(x^{k+1}) \preceq_C D^{k+1}.
\]

By (39), we have \( D^{k+1} = \eta_{k+1} D^k + (1 - \eta_{k+1}) F(x^{k+1}) \). Re-arranging and subtracting \( F(x^{k+1}) \) from both sides of this equality gives
\[
D^{k+1} - F(x^{k+1}) = \eta_{k+1}(D^k - F(x^{k+1})),
\]

\( \square \) Springer
which, combined with (40), yields that \( F(x^{k+1}) \leq C D^{k+1} \). This completes the proof.

In the next proposition, we prove that the CondG algorithm with the nonmonotone line search is well-defined, so that the iterates \( \{x^k\} \) can be generated.

**Proposition 10** Let \( x^k \) be an iterate of the CondG algorithm with the nonmonotone line search (38). If \( JF(x^k) \prec C 0 \), then there exists some \( \hat{t} \in (0, 1] \) such that

\[
F(x^k + t_d^k) \prec C D^k + \beta t_k JF(x^k) d^k, \quad \forall t_k \in (0, \hat{t}].
\]

**Proof** The proof follows from [13, Proposition 2.1] and Lemma 4.

**Lemma 5** Let \( \{x^k\} \) be the sequence generated by the CondG algorithm with the nonmonotone line search (38). Then, the following statements hold:

(i) The sequence \( \{D^k\} \) is monotone decreasing;

(ii) For all \( k \geq 0, x^k \in L \cap \Omega \), where \( L = \{x \in \mathbb{R}^n : F(x) \leq C F(x^0)\} \) and \( x^0 \in \Omega \) is a given initial point.

**Proof** Let us first prove item (i). According to (39) and (40), for all \( k \geq 0 \), we have

\[
D^{k+1} = \eta_{k+1} D^k + (1 - \eta_{k+1}) F(x^{k+1}) \leq C \eta_{k+1} D^k + (1 - \eta_{k+1}) D^k = D^k , \quad (41)
\]

which implies that item (i) holds.

We proceed item (ii) by induction. For \( k = 0, x^0 \in L \cap \Omega \) is clear. Assume that \( x^k \in L \) when \( k \leq l \), that is, \( F(x^k) \leq C F(x^0) \) for all \( k \leq l \). Note that, by Lemma 4 and (41), one has

\[
F(x^{l+1}) \leq C D^{l+1} \leq C D^l \leq \cdots \leq C D^0 = F(x^0),
\]

i.e., \( x^{k+1} \in L \). This, together with Remark 5, gives \( x^k \in L \cap \Omega \).

**Lemma 6** The sequence \( \{D^k\} \) is convergent, and

\[
\lim_{k \to \infty} \varphi_C (F(x^{k+1}) - D^k) = 0. \quad (42)
\]

**Proof** Since \( F \) is continuous, by Lemma 5(ii) and Lemma 4, there exists \( \tilde{F} \in \mathbb{R}^m \) such that \( \tilde{F} \leq C F(x^k) \leq C D^k \), i.e., \( \{D^k\} \) is bounded from below. Hence, it is convergent by Lemma 5(i). From the continuity of \( \varphi_C \), we immediately get \( \{\varphi_C (D^k)\} \) is convergent. Using (39) and (40), we have

\[
D^{k+1} - D^k = (1 - \eta_{k+1})(F(x^{k+1}) - D^k) \leq C (1 - \eta_{\text{max}})(F(x^{k+1}) - D^k) \leq C 0.
\]

(43)

By (43) and Lemma 1(iv)–(vi) and observing that \( \varphi_C (0) = 0 \), we obtain

\[
\varphi_C (D^{k+1}) - \varphi_C (D^k) \leq \varphi_C (D^{k+1} - D^k) \leq (1 - \eta_{\text{max}}) \varphi_C (F(x^{k+1}) - D^k) \leq 0.
\]
Therefore, (42) holds by taking \( \lim_{k \to \infty} \) on the above relation. \( \square \)

**Theorem 11** Let \( \{x^k\} \) be a sequence produced by the CondG algorithm with the nonmonotone line search (38). Then, every accumulation point of \( \{x^k\} \) is a stationary point of problem (1).

**Proof** Let \( \hat{x} \in \mathbb{R}^n \) be a accumulation point of the sequence \( \{x^k\} \). Clearly, the feasibility of \( \hat{x} \) follows from Remark 5 and the closedness of \( \Omega \). Then, there exists a subsequence \( \{x^{k_j}\}_j \) of \( \{x^k\} \) such that \( \lim_{j \to \infty} x^{k_j} = \hat{x} \). According to Proposition 2(ii), it is enough to see that \( v(\hat{x}) = 0 \). By (38) and Lemma 1(iv)–(vi), we have

\[
\varphi_c(F(x^k + t_k d^k) - D^k) \leq \beta t_k \varphi_c(JF(x^k)d^k) \quad (44)
\]

Using (42), (44) and the fact that \( JF(x^k)d^k \prec_C 0 \), as well as Lemma 1(ii), we get

\[
0 = \lim_{k \to \infty} \varphi_c(F(x^k + t_k d^k) - D^k) \leq \lim_{k \to \infty} \beta t_k \varphi_c(JF(x^k)d^k) \leq 0.
\]

Therefore,

\[
\lim_{k \to \infty} t_k v(x^k) = \lim_{k \to \infty} t_k \varphi_c(JF(x^k)d^k) = 0.
\]

We now consider the subsequence \( \{t_{k_j}\}_j \) of \( \{t_k\} \). Taking into consideration that \( t_k \in (0, 1] \) for all \( k \), we have the following two cases:

(a) \( \limsup_{j \to \infty} t_{k_j} > 0 \) or (b) \( \limsup_{j \to \infty} t_{k_j} = 0 \).

Repeating now the corresponding arguments in the proof of Theorem 6, we arrive at

\[
v(\hat{x}) \leq 0, \quad (45)
\]

and the subsequences \( \{x^{k_{ji}}\}_i, \{d(x^{k_{ji}})\}_i \) and \( \{t_{k_{ji}}\}_i \) converging to \( \hat{x}, d(\hat{x}) \) and 0, respectively. Take some fixed but arbitrary positive integer \( l \). Since \( \lim_{i \to \infty} t_{k_{ji}} = 0 \), we have \( t_{k_{ji}} < \delta^l \) for \( i \) large enough, which means that the nonmonotone line search (38) does not hold at \( x^{k_{ji}} \) for \( t = \delta^l \), i.e.,

\[
F(x^{k_{ji}} + \delta^l d(x^{k_{ji}})) \ngeq_C D^{k_{ji}} + \delta^l \beta JF(x^{k_{ji}})d(x^{k_{ji}}).
\]

Equivalently,

\[
F(x^{k_{ji}} + \delta^l d(x^{k_{ji}})) - D^{k_{ji}} - \delta^l \beta JF(x^{k_{ji}})d(x^{k_{ji}}) \notin -C,
\]

which means that

\[
F(x^{k_{ji}} + \delta^l d(x^{k_{ji}})) - D^{k_{ji}} - \delta^l \beta JF(x^{k_{ji}})d(x^{k_{ji}}) \in \mathbb{R}^m \setminus (-C)
\]

\[
= \text{int}(\mathbb{R}^m \setminus (-C)).
\]

\( \square \) Springer
By (46) and Lemma 1(ii), we have
\[
\varphi_C(F(x_{ji}^k + \delta d(x_{ji}^k)) - D^k_{ji} - \delta J F(x_{ji}^k)) > 0.
\] (47)

Since (47) holds for any positive integer \(l\), using Proposition 10 with \(k = k_{ji}\) and Lemma 1(vi), we conclude that \(J F(x_{ji}^k)d(x_{ji}^k) \not\in \mathbb{R}^m \setminus (-\text{int}(C))\). By Lemma 1(ii), it holds that
\[
\varphi_C(J F(x_{ji}^k)d(x_{ji}^k)) \geq 0.
\] (48)

Since \(F\) is continuously differentiable and \(\varphi_C\) is continuous, taking limits in (48) with \(i \rightarrow \infty\), we have \(v(\hat{x}) = \varphi_C(J F(\hat{x})d(\hat{x})) \geq 0\). This, combined with (45), yields that \(v(\hat{x}) = 0\).

5 Further discussions

In this section, we will further discuss several issues concerning the CondG algorithm, including the form of the subproblem (5) used to obtain a descent direction and the adaptive step size rule (28) used to get a step size.

From the CondG algorithm provided in Sect. 3, it is clear that the descent direction depends on the solution of the subproblem (5) at each iteration. However, the subproblem (5) seems mysterious and is not easy to implement from an intuitive point of view. The authors are indebted to the referee for giving some directions on the subproblem (5), which directly stimulates the forthcoming discussion. For any nonempty subset \(M\) of \(\mathbb{R}^m\), we denoted by
\[
S_M = \{\xi \in M : \|\xi\| = 1\}.
\]
The positive polar cone of \(C\) is denoted by
\[
C^* = \{\xi \in \mathbb{R}^m : \langle \xi, y \rangle \geq 0, \forall y \in C\}.
\]

By Corollary 2 in [53], we immediately have the following relation:
\[
\varphi_C(y) = \max_{\xi \in S_{C^*}} \langle \xi, y \rangle, \quad \forall y \in \mathbb{R}^m.
\]

Therefore, the subproblem (5) with \(x = x^k\) at \(k\)-iteration becomes the following general form:
\[
\min_{s \in \Omega} \max_{\xi \in S_{C^*}} \langle \xi, J F(x^k)(s - x^k) \rangle.
\] (49)
which is obviously a min-max optimization problem and has a bilinear objective function. Equivalently, (49) can be expressed as follows:

\[
\begin{aligned}
\min_{\gamma} & \\
\text{s.t.} & \langle \xi, JF(x^k)(s - x^k) \rangle \leq \gamma, & \forall \xi \in S_{C^*}, \\
& s \in \Omega,
\end{aligned}
\]

which can be treated as a semi-infinite optimization problem. Clearly, the specific form of (49) or (50) relies on the set \(S_{C^*}\) determined by the cone \(C\) and the norm \(\| \cdot \|\).

**Form 1.** Consider \(C = \mathbb{R}^m_+\) and the norm \(\| \cdot \|_{\infty}\). Then, we immediately obtain the following optimization problem:

\[
\begin{aligned}
\min & \gamma \\
\text{s.t.} & \langle \nabla f_i(x^k), s - x^k \rangle \leq \gamma, & \forall i \in I, \\
& s \in \Omega,
\end{aligned}
\]

which was considered in [49, 61].

**Form 2.** Consider \(C = \mathbb{R}^m_+\) and the norm \(\| \cdot \|_2\). Then, we get

\[
\begin{aligned}
\min & \gamma \\
\text{s.t.} & \sum_{i=1}^{m} \xi_i \langle \nabla f_i(x^k), s - x^k \rangle \leq \gamma, & \forall \xi \in M, \\
& s \in \Omega,
\end{aligned}
\]

where \(M = \{ \xi \in \mathbb{R}^m : \| \xi \|_2 = 1, \xi_i \geq 0, \forall i \in I \}\).

**Form 3.** Take \(m = 2\), the norm \(\| \cdot \|_{\infty}\) and two polyhedral cones

\[
C_1 = \left\{ y \in \mathbb{R}^2 : y_1 - y_2 \geq 0, y_1 + y_2 \geq 0 \right\},
\]

\[
C_2 = \left\{ y \in \mathbb{R}^2 : y_1 \geq 0, y_1 + y_2 \geq 0 \right\} \supseteq \mathbb{R}^2_+.
\]

Then, one has \(C_{1^*} = C_1\) and \(C_{2^*} = \{ y \in \mathbb{R}^2 : y_1 \geq y_2 \geq 0 \}\). Considering \(C_1\), we have

\[
\begin{aligned}
\min & \gamma \\
\text{s.t.} & \xi_1 \langle \nabla f_1(x^k), s - x^k \rangle + \xi_2 \langle \nabla f_2(x^k), s - x^k \rangle \leq \gamma, & \forall \xi \in \{1\} \times [-1, 1], \\
& s \in \Omega.
\end{aligned}
\]

\(\odot\) Springer
Considering $C_2$, we get

$$
\min \gamma \\
\text{s.t. } \xi_1 \langle \nabla f_1(x^k), s - x^k \rangle + \xi_2 \langle \nabla f_2(x^k), s - x^k \rangle \leq \gamma, \quad \forall \xi \in \{1\} \times [0, 1],
$$

(52)

Besides, we can also consider more subproblems by combining norms with polyhedral cones or non-polyhedral cones like the second-order cone $C = \{(x, y, z) \in \mathbb{R}^3 : x \geq \sqrt{y^2 + z^2}\}$ to construct the set $S_{C^*}$ and then derive the corresponding subproblems. Therefore, it is no longer a conceptual scheme for us to establish an intuitive and implementable subproblem by (49) or (50).

One of our purpose in this paper is to propose the adaptative step size and obtain the convergence result of the CondG algorithm with it from the theoretical side. The second issue concerns the choice of parameter $L$ in the adaptative step size, which ensures the $C$-convexity of $G$ on $\Omega$. For some simple functions, we can find $L$ that meets condition (A) by checking the $C$-convexity of $G$. Nevertheless, as in the Lipschitz constant $L_F$ of (21), this parameter $L$ is also not easily estimated for general functions. It is worth mentioning that Bauschke et al. [52] studied some problems with special structures like Poisson linear inverse problems and non-negative linear systems and then obtained the lower bound of parameter $L$ of the LC conditions in scalar optimization. Their work stimulates us to apply the CondG algorithm with adaptative step size to vector optimization problems with special structure so that we can adequately evaluate the parameter $L$ and recognize the potential of this step size rule. This will be an interesting work in future research.

6 Numerical experiments

This section is dedicated to the numerical experiments for verifying the algorithmic performance under two different settings, Form 1 and Form 3. We would like to emphasize that the present numerical experiments mainly focus on the practical behavior of the monotone and nonmonotone line searches under Form 1.

6.1 Numerical tests to Form 1

We first describe some preparations, including the comparison algorithms, test problems, performance profiles and performance metrics. Then we present and discuss the obtained results detailedly.

(a) Comparison algorithms

We compare the proposed method with some existing iterative algorithms that need to compute the descent direction in the literature. Here, we list the following methods in the tests reported:

- CGArm: the CondG algorithm with Armijo step size (12);
- CGNon: the CondG algorithm with nonmonotone linear search (38);
\begin{table}[h!]
\centering
\begin{tabular}{|c|c|c|c|c|c|c|c|c|c|c|c|}
\hline
Problem & Source & $m$ & $n$ & Problem & Source & $m$ & $n$ & Problem & Source & $m$ & $n$ \\
\hline
AP3 & [49] & 2 & 2 & JOS1 & [49] & 2 & 50 & PNR & [49] & 2 & 2 \\
Bel1 & [62] & 2 & 2 & & & & & QV1 & [49] & 2 & 10 \\
BNH & [63] & 2 & 2 & KW2 & [49] & 2 & 2 & SCH1 & [64] & 2 & 1 \\
Comet & [65] & 3 & 3 & LE1 & [49] & 2 & 2 & SLC2 & [66] & 2 & 10 \\
CONSTER & [63] & 2 & 2 & Lov2 & [49] & 2 & 2 & & & 2 & 100 \\
DTLZ1 & [65] & 3 & 10 & LTDZ & [49] & 3 & 3 & SP1 & [49] & 2 & 2 \\
Example2a & – & 2 & 1 & MMR1 & [49] & 2 & 2 & VU1 & [49] & 2 & 2 \\
FA1 & [49] & 3 & 3 & MMR3 & [49] & 2 & 2 & VU2 & [49] & 2 & 2 \\
Far1 & [49] & 2 & 2 & MMR4 & [49] & 2 & 2 & ZDT1 & [49] & 2 & 30 \\
FF1 & [49] & 2 & 2 & MOP3 & [49] & 2 & 2 & ZDT2 & [49] & 2 & 30 \\
IKK1 & [49] & 3 & 2 & MOP5 & [49] & 3 & 2 & ZDT3 & [49] & 2 & 30 \\
IM1 & [49] & 2 & 2 & MOP6 & [49] & 2 & 2 & ZDT4 & [49] & 2 & 30 \\
Kita & [63] & 2 & 2 & MOP7 & [49] & 3 & 2 & ZDT6 & [49] & 2 & 10 \\
\hline
\end{tabular}
\caption{List of test problems}
\end{table}

- PGArm: the projected gradient algorithm with Armijo step size (12) (see [14, Section 2.1]);
- PG Ave: the projected gradient algorithm with the vector version of average-type nonmonotone linear search (see [46, Algorithm 2.1]);
- PG Max: the projected gradient algorithm with the vector version of max-type nonmonotone linear search (see [47, Algorithm 1]).

(b) Test problems
Although the CondG method only enjoys global convergence under the convexity assumption, we also want to observe whether it performs well for non-convex problems. Therefore, we choose a set of problems, both convex and nonconvex, to make up the benchmark test problems in our empirical studies, which are summarized in Table 1. The first column contains the names of test problems. The second column shows the corresponding references. The dimensions of objectives and variables of these problems are given in the third and fourth columns, respectively. Note that Bel1, BNH, CONSTER and Kita are general constrained problems, and the rests are box-constrained problems. The lower and upper bounds of variables for SCH1 are set as $-3$ and $3$, respectively. The lower and upper bounds of variables for SLC2 are set as $-10$ and $10$, respectively. The bounds of the remaining problems are configured as described in the given sources.

(c) Implementation details
The experiments were performed by using MATLAB R2020b software on a PC with the specification: Processor Intel i7-10700 and 2.90 GHz, 32.00 GB RAM. We used the solver \texttt{fmincon} to solve the subproblem of the corresponding algorithm. The parameters used in CondG algorithm are $\beta = 10^{-4}$, $\delta = 0.5$ and $\eta_k$ is dynamically updated by the following formula: $\eta_k = 0.9/k$. For the termination criterion (Step 1 of CondG algorithm), we use $|v^k| \leq \epsilon$ with $\epsilon = 10^{-6}$. The maximum number of

\begin{quote}
\copyright Springer
\end{quote}
allowed outer iterations was set as 1000. The parameters used in PGArm algorithm are $\sigma = 10^{-4}$ and $\hat{\beta} = 1$. The parameters used in PGAve algorithm are $\sigma = 10^{-4}$, $\eta_{\min} = 0.1$, $\eta_{\max} = 0.9$ and $\beta_k = 1$. The parameters used in PGMax algorithm are $\sigma = 10^{-4}$, $\beta_k = 1$ and $M = 10$. For each test problem, we run the algorithm 100 times using initial points from a uniform random distribution belonging to the corresponding feasible region. Note that these algorithms enjoy the same initial points on each problem.

**d) Performance profiles**

To discuss the computational performance needed by an algorithm to solve a problem in scalar optimization, Dolan and Moré [67] proposed an efficient tool called performance profiles. It is worth mentioning that such a tool was also used in multiobjective optimization (see [12, 44, 47, 68]). Therefore, we describe further numerical results using this tool. For completeness, we briefly describe the tool in the sequel. Suppose that there exist $n_s$ solvers and $n_p$ problems. For each solver $s \in S$ and problem $p \in \mathcal{P}$, we denote $t_{p,s}$ by the performance (for example, the number of iterations) of the solver $s$ on the problem $p$. Define the performance ratio $z_{p,s} = t_{p,s} / \min \{t_{p,s} : s \in S\}$ and the cumulative distribution function $\rho : [1, \infty) \rightarrow [0, 1]$ by

$$
\rho(\tau) = \frac{1}{n_p} \left| \{ p \in \mathcal{P} : z_{p,s} \leq \tau \} \right|
$$

Then, the performance profiles are obtained by plotting the graph of the cumulative distribution function $\rho$. The value of $\rho(1)$ is the probability that the solver will win over the rest of the solvers. The robustness related to a solver can be obtained on the extreme right of the graph of $\rho$.

**e) Performance metrics**

To compare the algorithms with respect to their ability to properly generate Pareto frontiers, we use the purity and ($\Gamma$ and $\Delta$) spread metrics, which were presented in [68]. Let $F_{p,s}$ be a solution set found by solver $s \in S$ for problem $p \in \mathcal{P}$. Let $F_p$ be an approximation to the true Pareto front of problem $p$, calculated by first forming $\bigcup_{s \in S} F_{p,s}$ and then removing from this set any dominated points.

- **Purity metric.** This metric measures the number of nondominated points belonging to $F_p$ that a solver is able to compute. Given a solver $s \in S$ and a problem $p \in \mathcal{P}$, it is denoted by the ratio

$$
\bar{t}_{p,s} = \frac{|F_{p,s} \cap F_p|}{|F_p|}.
$$

In order to discuss the Purity metric using the performance profile, we let $t_{ps} = 1/\bar{t}_{ps}$. Therefore, lower values of $t_{ps}$ mean better performance. If $\bar{t}_{ps} = 0$, then we set $t_{ps} = \infty$, meaning that solver $s$ was unable to provide even a single nondominated point for problem $p$.

---

1 The performance profiles were in this paper generated using the MATLAB code perfprof.m freely available in the website https://github.com/higham/matlab-guide-3ed/blob/master/perfprof.m.
• **Spread** metrics. The $\Gamma$ spread metric is designed to measure the maximum size of the “holes” of an approximated Pareto front. The $\Delta$ spread metric indicates how well the points are distributed in an approximated Pareto front. Assume that an approximated Pareto front with $N$ points (i.e., $\{x^1, x^2, \ldots, x^N\}$) is obtained by a solver $s$ on a problem $p$, and that these points are sorted by increasing order for each objective $j \in I$. Let $x^0$ and $x^{N+1}$ be the extreme points obtained from $F_p$. Then, they are defined by

$$
\Gamma_{p,s} = \max_{j \in I} \max_{i \in \{0,1,\ldots,N\}} \delta_{i,j}
$$

and

$$
\Delta_{p,s} = \max_{j \in I} \left( \frac{\delta_{0,j} + \delta_{N,j} + \sum_{i=1}^{N} |\delta_{i,j} - \bar{\delta}_j|}{\delta_{0,j} + \delta_{N,j} + (N-1)\bar{\delta}_j} \right)
$$

where $\delta_{i,j} = |F_j(x^{i+1}) - F_j(x^i)|$ and $\bar{\delta}_j$ ($j \in I$) is the average of the distance $\delta_{i,j}, i = 1, 2, \ldots, N$. In the performance profile, we let $t_{p,s} = \Gamma_{p,s}$ or $t_{p,s} = \Delta_{p,s}$, depending on the metric considered. We refer the readers to [68] for further details and discussions about the purity and spread metrics.

(f) **Numerical results**

As described in the last subsection, we present our numerical results in the form of data and performance profiles. Table 2 reports the average number of iterations ($n_i$) and the average number of function evaluations ($n_f$) for every algorithm on each test problem.

Note that the average number of gradient evaluations is omitted here because it is similar to $n_i$. We now give an overview of the numerical results reported in this table. First, in all compared algorithms, CGNon achieves the best performance on 16 problems in terms of both $n_i$, followed by CGArm, PGMax, PGArm and PGAv. CGArm and CGNon achieve the best performance on 11 problems in terms of both $n_f$, followed by PGAv, PGArm and PGMax. When comparing CGNon to CGArm only, there are 20, 12 and 7 problems where CGNon is respectively better, identical and worse to CGArm in terms of $n_i$, and 19, 12 and 8 problems where CGNon is respectively better, identical and worse to CGArm in terms of $n_f$. This shows that the step size of the nonmonotone line search (38) tends in fact to be larger than that of the monotone line search (12). When comparing the algorithms with Armijo line search, there are 22 and 16 problems where CGArm is respectively better and worse to PGArm in terms of $n_i$ and $n_f$. When comparing CGNon to PGAv and PGMax, which have different nonmonotone line searches, there are

(i) 24 and 17 problems where CGNon is respectively superior to PGAv in terms of $n_i$ and $n_f$;

(ii) 26 and 22 problems where CGNon is respectively superior to PGMax in terms of $n_i$ and $n_f$.

To present the results more intuitively, we use Fig. 2 to give the performance profiles related to the average number of iterations and function evaluations. As we have seen,
| Problem     | CGArm | CGNon | PGArm | PGAve | PGMax |
|-------------|-------|-------|-------|-------|-------|
|             | $n_i/n_f$ | $n_i/n_f$ | $n_i/n_f$ | $n_i/n_f$ | $n_i/n_f$ |
| AP3         | 23    | 23    | 23    | 208.1/164.95 | 121.82/777.24 |
| Bell        | 1.2/4.94 | 1.52/3.28 | 1.2/4.94 | 1.52/3.28 | 1.2/4.94 |
| BSNH        | 109.92/521.9 | 34.9/10.87 | 34.9/10.87 | 34.9/10.87 | 34.9/10.87 |
| Comet       | 3.49/24 | 3.49/24 | 3.49/24 | 3.49/24 | 3.49/24 |
| CONSTER     | 341.66/1224.23 | 475.91/583.32 | 475.91/583.32 | 475.91/583.32 | 475.91/583.32 |
| DTLZ1       | 119.7/2096.94 | 61.87/834.72 | 61.87/834.72 | 61.87/834.72 | 61.87/834.72 |
| example2    | 1.52/3.12 | 3.35/8.53 | 3.35/8.53 | 3.35/8.53 | 3.35/8.53 |
| FA1         | 12.28/87.79 | 12.28/87.79 | 12.28/87.79 | 12.28/87.79 | 12.28/87.79 |
| Far1        | 1.28/109.88 | 12.28/87.79 | 12.28/87.79 | 12.28/87.79 | 12.28/87.79 |
| FF1         | 50.54/722.11 | 50.54/722.11 | 50.54/722.11 | 50.54/722.11 | 50.54/722.11 |
| F1          | 27.14/105.47 | 27.14/105.47 | 27.14/105.47 | 27.14/105.47 | 27.14/105.47 |
| example2    | 1.52/3.12 | 1.52/3.12 | 1.52/3.12 | 1.52/3.12 | 1.52/3.12 |
| IKK1        | 341.66/1224.23 | 475.91/583.32 | 475.91/583.32 | 475.91/583.32 | 475.91/583.32 |
| IMI         | 23.18/5.36 | 23.18/5.36 | 23.18/5.36 | 23.18/5.36 | 23.18/5.36 |
| JOSI        | 230.59/465.98 | 230.59/465.98 | 230.59/465.98 | 230.59/465.98 | 230.59/465.98 |
| Kita        | 141.86/592.72 | 141.86/592.72 | 141.86/592.72 | 141.86/592.72 | 141.86/592.72 |
| KV2         | 76.08/220.24 | 76.08/220.24 | 76.08/220.24 | 76.08/220.24 | 76.08/220.24 |
| LE1         | 18.01/207.29 | 18.01/207.29 | 18.01/207.29 | 18.01/207.29 | 18.01/207.29 |
| Lov2        | 3.37/12.89 | 3.37/12.89 | 3.37/12.89 | 3.37/12.89 | 3.37/12.89 |
| Problem | CGArm \( n_i/n_f \) | CGNon \( n_i/n_f \) | PGArm \( n_i/n_f \) | PGAve \( n_i/n_f \) | PGMax \( n_i/n_f \) |
|--------|-----------------|-----------------|-----------------|-----------------|-----------------|
| LTDZ   | 2/3             | 2/3             | 2.15/3.3        | 2.16/3.32       | 6.88/173.24     |
| MMR1   | 177.29/6162.7   | 171.42/5819.78  | 245.8/8519.61   | 246.31/8602.6   | 235.74/8145.02  |
| MMR3   | 2/3             | 2/3             | 9.35/17.7       | 9.18/17.36      | 15.94/257.66    |
| MMR4   | 2.03/3.06       | 2.03/3.06       | 5.22/9.44       | 4.84/8.68       | 13.49/320.08    |
| MOP3   | 34.3/1024.43    | 8.37/72.4       | 4.47/11.84      | 8.62/20.14      | 20.2/293.16     |
| MOP5   | 1.78/10.87      | 1.73/9.88       | 1.86/2.72       | 1.92/8.92       | 2.07/8.92       |
| MOP6   | 2.72/6.59       | 2.39/4.58       | 2.74/6.16       | 2.54/4.94       | 5.49/118.68     |
| MOP7   | 10.33/86.99     | 14.34/90.65     | 120.43/239.86   | 132.77/264.54   | 137.86/446.42   |
| PNR    | 22.3/329.5      | 21.81/280.72    | 7.37/35.61      | 5.18/19.16      | 9.03/65.95      |
| QV1    | 444.92/6665.38  | 351.27/5471.24  | 535.15/1074.12  | 523.12/1046     | 526.96/1200.35  |
| SCH1   | 1.62/3.48       | 1.62/3.48       | 1.62/2.86       | 1.62/2.86       | 2.97/47.18      |
| SLC2   | 111.59/2962.94  | 37.65/478.71    | 14.43/54.41     | 20.51/59.51     | 32.57/122.86    |
| SP1    | 78.28/1996.05   | 38.3/516.73     | 15.07/48.3      | 26.24/74.36     | 40.74/143.99    |
| VU1    | 100.98/3290.79  | 78.65/2185.55   | 12.63/41.76     | 39.02/121.19    | 747.3/2280.38   |
| RU2    | 253.43/3786.49  | 252.6/3779.61   | 117.56/415.59   | 122.83/414.39   | 105.81/607.09   |
| ZDT1   | 2.28/3.9        | 2.28/3.9        | 4.22/7.4        | 4.2/7.4         | 7.78/136.28     |
| ZDT2   | 451.61/4099.75  | 3695.65         | 587.83/184.5    | 506.58/4412.39  | 77.86/798.61    |
| ZDT3   | 2/3             | 3/2            | 2.35/3.7        | 2.33/6.66       | 7.94/205.62     |
| ZDT4   | 3.78/12.5       | 3.09/7.51      | 25.65/76.91     | 16.31/40.84     | 14.59/200.19    |
| ZDT6   | 13.31/175.41    | 17/211.11      | 11.4/86.72      | 26.04/213.52    | 7.28/74.43      |
the CGNon takes fewer iterations to find stationary points than its competitors on these problems when $\tau \geq 2.4$, while it consumes more function evaluations than PGArm and PGAve when $\tau \geq 1.6$. We can also observe that CGNon performs better than CGArm, which is attributed to the nonmonotone line search.

Until now, we are just concerned with the speed of convergence of the methods. In what follows, it is interesting to see if the methods can generate the Pareto front properly. For this, we choose four problems (BNH, Comet, KW2 and SCH1) from Table 1, plotting the stationary points found by these algorithms. As seen from Fig. 3, the five algorithms can get desirable solutions on BNH, Comet and SCH1. This is because these algorithms can converge to weakly efficient solutions for convex problems. But for nonconvex KW2, the solutions found by these algorithms do not seem good enough.

We further use the performance metrics to present comparisons of these algorithms. The first measure to be considered is the purity. As recommended in [68], we compare the algorithms in pairs when using it. From Fig. 4, we can observe that CGNon performs better than CGArm on our test problems. One reasonable explanation for this occurrence is that the CondG algorithm with the nonmonotone line search (38) can find better stationary points closer to the Pareto front. It can be easily seen from 2 to 7 pictures in Fig. 4 that PGArm, PGAve and PGMax have inferior performance.

For a visual comparison, Fig. 5 plot the final solutions obtained by these algorithms on Lov2, MMR3, ZDT3 and ZDT6. Clearly, from Figs. 3, 4, 5, CGNon is better on KW2, Lov2 and ZDT6 than CGArm because CGNon can achieve more nondominated points. CGArm and CGNon significantly outperform PGArm, PGAve and PGMax on KW2, Lov2 and ZDT3. For MMR3 and ZDT6, PGAve and PGMax equipped with nonmonotone line search perform better than PGArm.

The final step of our numerical results compares the spread metric (i.e., $\Gamma$ and $\Delta$), which are depicted in Fig. 6. For the $\Gamma$ metric, one can observe that CGArm exhibits the best overall performance and that CGNon is slightly more efficient than PGArm and PGAve. For the $\Delta$ metric, PGMax does not seem good enough, and no significant difference among the rest algorithms is noticed.
Fig. 3 The final solutions found by CGArm, CGNon, PGAve and PGM in BNH, Comet, KW2 and SCH1
Fig. 4 Performance profile using the purity metric

Fig. 5 The final solutions found by CGArm, CGNon, PGapm, PGapve and PGapmax on Lov2, MMR3, ZDT3 and ZDT6
6.2 A preliminary attempt to Form 3

This section presents two vector optimization problems to illustrate how the proposed CondG algorithm with Armijo, adaptative and nonmonotone step size rules work under Form 3. To the best of our knowledge, this is the first attempt to directly solve vector optimization problems with an iterative algorithm in the case of $C \neq \mathbb{R}^m_+$. 

Problem 1 The first problem we construct has the following form

$$\min_C F(x) = (x + 1, x^2 - x - 2)^T$$

s.t. $x \in [0, 1]$.

Problem 2 The second problem is the following

$$\min_C F(x) = \left( \frac{1}{n} \sum_{i=1}^{n} x_i^2, \frac{1}{n} \sum_{i=1}^{n} (x_i - 2)^2 \right)^T$$

s.t. $x \in [-2, 2]$,

where the objective functions are selected from JOS1. In the sequel, we set $n = 5$.

Clearly, the two problems are $\mathbb{R}^2_+$-convex on the domains. Problems 1 and 2 are $C_2$-convex but not $C_1$-convex. We discrete the domains and plot all the image points for the two problems. As shown in Figs. 7 and 8, the black points stand for the image points of problems 1 and 2. Thus, they provide good representations of the image space of $F$. To intuitively show the image set of solution sets with respect to the cones $\mathbb{R}^2_+$, $C_1$ and $C_2$, we have marked them in green, red and cyan in Figs. 7 and 8, respectively. It came as no surprise to see they have different shapes.

To solve the subproblems (51) and (52), we used the standard MATLAB subroutine fseminf. In our experiments, we selected 40 random initial points from the domain. The intrinsic parameters for the CondG algorithm with Armijo and nonmonotone step size rules are set the same as in Sect. 6.1(c). For the adaptative step size, we choose $e = (1, 0.5)^T$ and $L = 2$ for the two problems, which make the condition (A) holds.
The computational results obtained by the CondG algorithm under Form 1 and Form 3 are summarized in Table 3. In this table, $n_i$ and $n_f$ stand, respectively, for the average number of iterations and the average number of function evaluations, used by each algorithm. Since CondG algorithm with adaptative step size (CGAda) does not need to evaluate the function values, we use the symbol “−” in this table.

From Table 3, the following observations can be made. First, CGAda needs more iterations and function evaluations than CGArm and CGNon. Second, the performance of CGArm and CGNon is consistent on problem 1. This is probably a consequence of the simplicity of this problem. Third, under Form 3, CGNon performs slightly better than CGArm on problem 2 in terms of the values of $n_i$ and $n_f$, the opposite observation is shown under Form 1.

In order to observe how the iteration points move under Form 1 and Form 3, we depict the trajectories of iteration points, as shown in Figs. 9, 10, 11. In these figures, dashed lines corresponding to the path of iterations of the algorithms, snowflake points corresponding to the initial points, and the circles corresponding to the solutions found by the algorithms. By comparing Figs. 9, 10, 11, we can clearly see that the motion paths of iteration points found by the algorithms are slightly different under different cones. From Figs. 10 and 11, it is observed that, under Form 3, the CondG algorithm is able to identify an approximation of the solution set for the two problems.
Table 3  Average number of iterations and function evaluations for CGArm, CGAda and CGNon on problems 1 and 2

| Problem | Form 1 CGArm $n_i/n_f$ | CGAda $n_i/n_f$ | CGNon $n_i/n_f$ | Form 3 ($C = C_1$) CGArm $n_i/n_f$ | CGAda $n_i/n_f$ | CGNon $n_i/n_f$ | Form 3 ($C = C_2$) CGArm $n_i/n_f$ | CGAda $n_i/n_f$ | CGNon $n_i/n_f$ |
|---------|----------------------|----------------|----------------|-------------------------------|----------------|----------------|-------------------------------|----------------|----------------|
| 1       | 0.53/2.73            | 3.2/-         | 0.53/2.73      | 1/3                           | 51.27/-        | 1/3            | 1/3                           | 51.27/-        | 1/3            |
| 2       | 26.82/311.68         | 233.73/-      | 35.53/370.53   | 18.13/183.8                   | 59.58/-        | 17/187.3       | 38.38/873.8                  | 133.93/-       | 36.02/718.18   |

Fig. 9 The final solutions found by CGArm, CGAda and CGNon on problems 1 and 2 under Form 1.

Fig. 10 The final solutions found by CGArm, CGAda and CGNon on problems 1 and 2 under Form 3 with $C = C_1$. 
Conclusions

In this work, we have proposed a conditional gradient method for solving constrained vector optimization problems. In our method, the auxiliary subproblem used to obtain the descent direction is constructed based on the well-known Hiriart-Urruty’s oriented distance function, and the step sizes are found by Armijo, adaptive and nonmonotone strategies. We proved the sequence generated by the method can converge to a stationary point no matter how bad is our initial point. Moreover, under \( C \)-convexity assumptions of the objective function, the accumulation point of this subsequence is a weakly efficient solution. When \( C = {\mathbb{R}}_+^m \), preliminary numerical experiences on benchmark problems indicate that the proposed method with nonmonotone line search is efficient. We also tried to solve two problems in the case of a given cone \( C \) instead of \( {\mathbb{R}}_+^m \) in Sect. 6.2. Numerical results indicate that our attempt is successful.

Let us mention some remaining problems deserving further investigation, which are raised one by one below. Firstly, it might be interesting to study the open question already mentioned naturally in Remark 8. Secondly, we will continue to explore the works related to the parameter \( L \) mentioned in Section 5. Our numerical experiment in the setting of a polynomial cone rather than non-negative orthant is just a preliminary trial. It seems to be a relevant topic for future research to verify the performance of the CondG algorithm on more vector optimization problems under the case of Form 3 or other forms. We leave addressing this issue as our third research. Finally, a future path we consider applying the proposed nonmonotone line search (38) to other methods like the steepest descent [7], trust region [12] and projected gradient [14] methods.
Acknowledgements We thank the anonymous referees for their constructive comments and suggestions to improve the quality and contributions of the paper. Also, we would like to thank Prof. L.P. Tang for the helpful suggestions and discussions during the writing of this paper.

Author contributions WC conceived of the study and drafted the manuscript. XY participated in its analysis and coordination and helped to draft the manuscript. YZ participated in its analysis and was responsible for checking the content and grammar of the article.

Funding This research was supported by the Major Program of the National Natural Science Foundation of China (11991020, 11991024), the National Natural Science Foundation of China (11971084, 12001072, 12271067), NSFC-RGC (Hong Kong) Joint Research Program (12261160365), the Team Project of Innovation Leading Talent in Chongqing (CQYC20210309536), the Natural Science Foundation of Chongqing (CSTB2022N5CQ-MSX1318, csc2019jcyj-zdxmX0016), the China Postdoctoral Science Foundation Project (2019M653332).

Data availability No data and materials.

Code Availability No code.

Declarations

Conflict of interest The authors declare no competing financial or non-financial interests.

References

1. John, J.: Vector Optimization: Theory, Applications and Extensions, 2nd edn. Springer, Berlin (2011)
2. Rangaiah, G.P., Bonilla-Petriciolet, A.: Multi-Objective Optimization in Chemical Engineering: Developments and Applications. Wiley (2013)
3. Zopounidis, C., Galariotis, E., Dounpos, M., Sarri, S., Andriosopoulos, K.: Multiple criteria decision aiding for finance: an updated bibliographic survey. Eur. J. Oper. Res. 247(2), 339–348 (2015)
4. Fliege, J.: OLAF-a general modeling system to evaluate and optimize the location of an air polluting facility. OR Spektrum. 23(1), 117–136 (2001)
5. Tavana, M., Sodenkamp, M.A., Suhl, L.: A soft multi-criteria decision analysis model with application to the European Union enlargement. Ann. Oper. Res. 181(1), 393–421 (2010)
6. Jin, Y.C.: Multi-Objective Machine Learning. Springer-Verlag, Berlin (2006)
7. Fliege, J., Svaiter, B.F.: Steepest descent methods for multicriteria optimization. Math. Methods Oper. Res. 51(3), 479–494 (2000)
8. Fliege, J., Graña Drummond, L.M., Svaiter, B.F.: Newton’s method for multiobjective optimization. SIAM J. Optim. 20(2), 602–626 (2009)
9. Wang, J., Hu, Y., Wai Yu, C.K., Li, C., Yang, X.: Extended Newton methods for multiobjective optimization: majorizing function technique and convergence analysis. SIAM J. Optim. 29(3), 2388–2421 (2019)
10. Da Cruz Neto, J.X., Da Silva, G.J.P., Ferreira, O.P., Lopes, J.O.: A subgradient method for multiobjective optimization. Comput. Optim. Appl. 54(3), 461–472 (2013)
11. Qu, S.J., Goh, M., Chan, F.T.S.: Quasi-Newton methods for solving multiobjective optimization. Oper. Res. Lett. 39(5), 397–399 (2011)
12. Carrizo, G.A., Lotito, P.A., Maciel, M.C.: Trust region globalization strategy for the nonconvex unconstrained multiobjective optimization problem. Math. Program. 159(1), 339–369 (2016)
13. Graña Drummond, L.M., Svaiter, B.F.: A steepest descent method for vector optimization problems. J. Comput. Appl. Math. 175(2), 395–414 (2005)
14. Graña Drummond, L.M., Iusem, A.N.: A projected gradient method for vector optimization problems. Comput. Optim. Appl. 28(1), 5–29 (2004)
15. Fukuda, E.H., Graña Drummond, L.M.: Inexact projected gradient method for vector optimization. Comput. Optim. Appl. 54(3), 473–493 (2013)
16. Aliprantis, C.D., Florenzano, M., Martins da Rocha, V.F., Tourky, R.: Equilibrium analysis in financial markets with countably many securities. J. Math. Econom. 40(6), 683–699 (2004)
17. Aliprantis, C.D., Florenzano, M., Tourky, R.: General equilibrium analysis in ordered topological vector spaces. J. Math. Econom. 40(3–4), 247–269 (2004)
18. Gutiérrez, C., Jiménez, B., Novo, V.: On approximate solutions in vector optimization problems via scalarization. Comput. Optim. Appl. 35(3), 305–324 (2006)
19. Ansari, Q.H., Köbis, E., Yao, J.C.: Vector Variational Inequalities and Vector Optimization. Springer International Publishing AG, Cham (2018)
20. Villacorta, K.D.V., Oliveira, P.R.: An interior proximal method in vector optimization. Eur. J. Oper. Res. 214(3), 485–492 (2011)
21. Chen, Z.: Generalized viscosity approximation methods in multiobjective optimization problems. Comput. Optim. Appl. 49(1), 179–192 (2011)
22. Bello Cruz, J.Y.: A subgradient method for vector optimization problems. SIAM J. Optim. 23(4), 2169–2182 (2013)
23. Lucambio Pérez, L.R., Prudente, L.F.: Nonlinear conjugate gradient methods for vector optimization. SIAM J. Optim. 28(3), 2690–2720 (2018)
24. Gonçalves, M.L.N., Prudente, L.F.: On the extension of the Hager-Zhang conjugate gradient method for vector optimization. Comput. Optim. Appl. 76(3), 889–916 (2020)
25. Fletcher, R., Reeves, C.M.: Function minimization by conjugate gradients. Comput. Optim. Appl. 10(1), 177–182 (1999)
26. Fletcher, R.: Unconstrained Optimization, Pract. Methods Optim. 1, Wiley, New York (1980)
27. Dai, Y.H., Yuan, Y.X.: A nonlinear conjugate gradient method with a strong global convergence property. SIAM J. Optim. 10(1), 177–182 (1999)
28. Polak, E., Ribiere, G.: Note sur la convergence de méthodes de directions conjuguées. Rev. Française Inform. Rech. Opér Sér Rouge. 3(16), 35–43 (1969)
29. Polyak, B.T.: The conjugate gradient method in extremal problems. USSR Comput. Math. Math. Phys. 9(4), 94–112 (1969)
30. Hestenes, M.R., Stiefel, E.: Methods of conjugate gradients for solving linear systems. J. Res. Nat. Bureau Standards 49(6), 409–436 (1952)
31. Hager, W.W., Zhang, H.C.: A new conjugate gradient method with guaranteed descent and an efficient line search. SIAM J. Optim. 16(1), 170–192 (2005)
32. Fukuda, E.H., Graña Drummond, L.M.: A survey on multiobjective descent methods. Pesquisa Oper. 34(3), 585–620 (2014)
33. Graña Drummond, L.M., Raupp, F.M.P., Svaiter, B.F.: A quadratically convergent Newton method for vector optimization. Optimization 63(5), 661–677 (2014)
34. Lu, F., Chen, C.R.: Newton-like methods for solving vector optimization problems. Appl. Anal. 93(8), 1567–1586 (2014)
35. Chen, Z., Huang, H.Q., Zhao, K.Q.: Approximate generalized proximal-type method for convex vector optimization problem in Banach spaces. Comput. Math. Appl. 57(7), 1196–1203 (2009)
36. Chuong, T.D., Yao, J.C.: Steepest descent methods for critical points in vector optimization problems. Appl. Anal. 91(10), 1811–1829 (2012)
37. Bonnel, H., Iusem, A.N., Svaiter, B.F.: Proximal methods in vector optimization. SIAM J. Optim. 15(4), 953–970 (2005)
38. Chuong, T.D.: Newton-like for efficient solutions in vector optimization. Comput. Optim. Appl. 54(3), 495–516 (2013)
39. Boţ, R.I., Grad, S.-M.: Inertial forward backward methods for solving vector optimization problems. Optimization 67(7), 959–974 (2018)
40. Grippo, L., Lampariello, F., Lucidi, S.: A nonmonotone line search technique for Newton’s method. SIAM J. Numer. Anal. 23(4), 707–716 (1986)
41. Zhang, H., Hager, W.W.: A nonmonotone line search technique and its application to unconstrained optimization. SIAM J. Optim. 14(4), 1043–1056 (2004)
42. Gu, N.Z., Mo, J.T.: Incorporating nonmonotone strategies into the trust region method for unconstrained optimization. Comput. Math. Appl. 55(9), 2158–2172 (2008)
43. Ahookhosh, M., Ghaderi, S.: On efficiency of nonmonotone Armijo-type line searches. Appl. Math. Model. 43, 170–190 (2017)
44. Mita, K., Fukuda, E.H., Yamashita, N.: Nonmonotone line searches for unconstrained multiobjective optimization problems. J. Global Optim. 75(1), 63–90 (2019)
45. Mahdavi-Amiri, N., Salehi Sadaghiani, F.: A superlinearly convergent nonmonotone quasi-Newton method for unconstrained multiobjective optimization. Optim. Methods Softw. 35(6), 1223–1247 (2020)

46. Fazzio, N.S., Schuverdt, M.L.: Convergence analysis of a nonmonotone projected gradient method for multiobjective optimization problems. Optim. Lett. 13(6), 1365–1379 (2019)

47. Carrizo, G.A., Fazzio, N.S., Schuverdt, M.L.: A nonmonotone projected gradient method for multiobjective problems on convex sets. J. Oper. Res. Soc. China. (2022). https://doi.org/10.1007/s40305-022-00410-y

48. Ramirez, V.A., Sottosanto, G.N.: Nonmonotone trust region algorithm for solving the unconstrained multiobjective optimization problems. Comput. Optim. Appl. 81(3), 769–788 (2022)

49. Assunção, P.B., Ferreira, O.P., Prudente, L.F.: Conditional gradient method for multiobjective optimization. Comput. Optim. Appl. 78(3), 741–768 (2021)

50. Hiriart-Urruty, J.B.: Tangent cone, generalized gradients and mathematical programming in Banach spaces. Math. Oper. Res. 4(1), 79–97 (1979)

51. Zaffaroni, A.: Degrees of efficiency and degrees of minimality. SIAM J. Control. Optim. 42(3), 1071–1086 (2003)

52. Bauschke, H.H., Bolte, J., Teboulle, M.: A descent lemma beyond Lipschitz gradient continuity: first-order methods revisited and applications. Math. Oper. Res. 42(2), 330–348 (2017)

53. Liu, C.G., Ng, K.F., Yang, W.H.: Merit functions in vector optimization. Math. Program. Ser. A. 119(2), 215–237 (2009)

54. Gao, Y., Hou, S.H., Yang, X.M.: Existence and optimality conditions for approximate solutions to vector optimization problems. J. Optim. Theory Appl. 152(1), 97–120 (2012)

55. Zhou, Z.A., Chen, W., Yang, X.M.: Scalarizations and optimality of constrained set-valued optimization using improvement sets and image space analysis. J. Optim. Theory Appl. 183(3), 944–962 (2019)

56. Ansari, Q.H., Köbis, E., Sharma, P.K.: Some properties of generalized oriented distance function and their applications to set optimization problems. J. Optim. Theory Appl. 193(1–3), 247–279 (2022)

57. Ansari, Q.H., Sharma P.K.: Set Order Relations, Set Optimization, and Ekeland’s Variational Principle, in Optimization, Variational Analysis and Applications, Edited by V. Laha, P. Maréchal and S. K. Mishra, Springer Proceedings in Mathematics and Statistics 355, Springer Nature Singapore Pvt. Ltd., pp. 103–165, (2021)

58. Bolte, J., Sabach, S., Teboulle, M., Vaisbourd, Y.: First order methods beyond convexity and Lipschitz gradient continuity with applications to quadratic inverse problems. SIAM J. Optim. 28(3), 2131–2151 (2018)

59. Dennis, J.E., Schnabel, R.B.: Numerical Methods for Unconstrained Optimization and Nonlinear Equations. Society for Industrial and Applied Mathematics, Philadelphia (1996)

60. Cocchi, G., Liuzzi, G., Papini, A., Sciandrone, M.: An implicit filtering algorithm for derivative-free multiobjective optimization with box constraints. Comput. Optim. Appl. 69(2), 267–296 (2018)

61. Belegundu, A.D., Murthy, D.V., Salagame, R.R., Constants, E.W.: Multi-objective optimization of laminated ceramic composites using genetic algorithms. In: Proc. 5th AIAA/NASA/USAF/ISSMO Symp. Multidisciplinary Analysis and Optimization, 1015–1022 (1994)

62. Tharwat, A., Houssein, E.H., Ahmed, M.M., Hassanien, A.E., Gabel, T.: MOGOA algorithm for constrained and unconstrained multi-objective optimization problems. Appl. Intell. 48(8), 2268–2283 (2018)

63. Srinivas, N., Deb, K.: Multiobjective optimization using nondominated sorting in genetic algorithms. Evol. Comput. 2(3), 221–248 (1994)

64. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable Test Problems for Evolutionary Multi-Objective Optimization. In: Abraham, A., Jain, R., Goldberg, R. (eds.) Evolutionary Multiobjective Optimization: Theoretical Advances and Applications, chapter 6, pp. 105–145. Springer (2005)

65. Schütze, O., Lara, A., Coello, C.A.: The directed search method for unconstrained multi-objective optimization problems. Technical report TR-OS-2010-01, http://delta.cs.cinvestav.mx/~schuetze/technical_20reports/TR-OS-2010-01.pdf.gz, (2010)

66. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. 91(2), 201–213 (2002)

67. Custódio, A.L., Madeira, J.A., Vaz, A.I.F., Vicente, L.N.: Direct multisearch for multiobjective optimization. SIAM J. Optim. 21(3), 1109–1140 (2011)
Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.