Frank-Wolfe method for vector optimization with a portfolio optimization application

W. Chen\textsuperscript{a}, X.M. Yang\textsuperscript{b} and Y. Zhao\textsuperscript{c}

\textsuperscript{a}College of Mathematics, Sichuan University, Chengdu, China; \textsuperscript{b}School of Mathematical Sciences, Chongqing Normal University, Chongqing, China; \textsuperscript{c}College of Mathematics and Statistics, Chongqing Jiaotong University, Chongqing, China

ARTICLE HISTORY
Compiled September 24, 2021

ABSTRACT
In this paper, we propose an extension of the classical Frank-Wolfe method for solving constrained vector optimization problems with respect to a partial order induced by a closed, convex and pointed cone with nonempty interior. In the proposed method, the construction of auxiliary subproblem is based on the well-known oriented distance function. Two types of stepsize strategies including Armijo line search and adaptive stepsize are used. It is shown that every accumulation point of the generated sequences satisfies the first-order necessary optimality condition. Moreover, under suitable convexity assumptions for the objective function, it is proved that all accumulation points of any generated sequences are weakly efficient points. We finally apply the proposed algorithms to a portfolio optimization problem under bicriteria considerations.

KEYWORDS
vector optimization; Frank-Wolfe method; stationary point; convergence; portfolio optimization

1. Introduction

Vector optimization problems arise, for example, in functional analysis, multiobjective programming, multicriteria decision making, statistics, approximation theory, cooperative game theory, etc (see [1]). In this class of problems, we seek to minimize several objectives with respect to a partial order induced by a closed, convex and pointed cone $C$ with nonempty interior.

A particular case of such problems, very important in the practical applications, is when $C := \mathbb{R}^m_+$, where $\mathbb{R}^m_+$ is the nonnegative orthant of $\mathbb{R}^m$. This case corresponds to the multicriteria or multiobjective optimization. To solve multiobjective optimization problems, one of the most popular strategy is the so-called scalarization method whose core idea is to convert a target multiobjective optimization problem into a scalar optimization problem (see [23]) and then solve this transformed scalar optimization problem by virtue of some classical optimization methods. However, the main disadvantage of scalarization methods is that it needs to introduce some additional and appropriate parameters in the transformation process, and this requires insight into
the problem structure which may not be available in general. Another new type of strategy is to extend classical optimization methods to multiobjective versions. For example, Fliege and Svaiter [4] proposed a suitable extension of the classical steepest descent method for multiobjective optimization. There are two key features of their method at each iteration: (i) a descent direction is obtained by solving a auxiliary and non-parametric quadratic scalar subproblem; (ii) Armijo line search is used to find a point that dominates the current one along this direction. Following the research works of Fliege and Svaiter [4], in recent years, several classical numerical iterative methods (e.g. Newton method, quasi-Newton method, projected gradient method, proximal gradient method, trust region method, conditional gradient method, etc.) for solving scalar optimization problems have been extended to solve multiobjective optimization problems (see for example [5–11] and references therein). Note that, in [8], the authors presented a rigorous and comprehensive survey on multiobjective versions of the steepest descent method, the projected gradient method and the Newton method. Compared with these methods summarized in [8], the conditional gradient method presented in [11] for constrained multiobjective optimization problems just need solve a linear subproblem over a compact convex set at every iteration.

To extend the methods for multiobjective optimization presented in [8], the authors [12–17], in finite-dimensional space, gave respectively the extensions of steepest descent method, projected gradient method and Newton method to solve vector optimization problems with respect to the general partial order rather than the nonnegative orthant. It is noteworthy that the subproblems given in [13–17] have more general forms, and their constructions are based on the well-known gauge function. In addition, by virtue of the general order cone \(C\), vector versions of the proximal point method [18], the nonmonotone gradient algorithm [19] and the Hager-Zhang conjugate gradient method [20] are introduced to solve vector optimization problems. In infinite-dimensional settings, there are also several methods for solving vector optimization problems (see [21–27] and references therein). For example, Chuong and Yao [25] presented exact and inexact steepest descent methods of vector optimization problems for a map from a finite dimensional Hilbert space to a Banach space, which generalizes the works in [4]. Very recently, Bot and Grad [27] have proposed two forward-backward proximal point type algorithms with inertial/memory effects for finding weakly efficient solutions to a vector optimization problem.

The goal of this paper is to present a new method for vector optimization problems. In this setting, the partial order is induced by a closed, convex and pointed cone \(C\) with nonempty interior in finite-dimensional space. Our method is consistent with the idea of the methods presented in [13–17] in that we seek to extend the Frank-Wolfe method for scalar optimization to vector optimization. At each iteration of our method, the descent direction is the difference between the previous iteration point and a optimal solution of a auxiliary subproblem defined by the well-known oriented distance function. Meanwhile, we consider two strategies of stepsizes: Armijo line search and adaptive stepsize. Under some reasonable conditions including vector version of descent lemma and \(C\)-boundedness, we establish the convergence results for Frank-Wolfe method with two different strategies of stepsizes, that is, the stationarity of accumulation points of the sequences generated by our method. Finally, we apply the method to bicriteria portfolio optimization problem so as to produce a optimal portfolio strategy for investors.

The outline of this paper is as follows. Section 2 presents some preliminaries on the notations. Section 3 explains the vector optimization problem and a necessary condition for optimality. In Section 4, the Frank-Wolfe method with Armijo line search
(see Algorithm 1) and adaptive stepsize (see Algorithm 3) for vector optimization problems are introduced and the convergence results of the produced sequences are obtained. An application to a bicriteria portfolio optimization problem is presented in Section 5. Finally, in Section 6, we make some conclusions about our works.

2. Preliminaries

For a nonempty set $X \subset \mathbb{R}^m$, the interior and boundary of $X$ are respectively denoted by $\text{int}(X)$ and $\text{bd}(X)$. Let $C \subset \mathbb{R}^m$ be a closed, convex and pointed cone with nonempty interior. For any $y_1, y_2 \in \mathbb{R}^m$, the partial order $\preceq$ in $\mathbb{R}^m$ induced by $C$ is defined as

$$y_1 \preceq y_2 \iff y_2 - y_1 \in C,$$

and the partial order $\prec$ in $\mathbb{R}^m$ induced by $\text{int}(C)$ is defined as

$$y_1 \prec y_2 \iff y_2 - y_1 \in \text{int}(C).$$

We now recall the concept of oriented distance function (also called assigned distance function or Hiriart-Urruty function), which was proposed by Hiriart-Urruty [28] 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 [29,30], optimality conditions for vector optimization [31], optimality conditions for set-valued optimization problems [32], etc. Herein, we consider the oriented distance function in $\mathbb{R}^m$.

**Definition 2.1.** [28] Let $A$ be a subset of $\mathbb{R}^m$. The function $\Delta_A : \mathbb{R}^m \to \mathbb{R} \cup \{±\infty\}$, 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 $A$ and $\|\cdot\|$ denotes the norm in $\mathbb{R}^m$.

Note that

$$\Delta_A(y) = \begin{cases} 
    d_A(y), & \text{if } y \in \mathbb{R}^m \setminus A, \\
    -d_{\mathbb{R}^m \setminus A}(y), & \text{if } y \in A.
\end{cases}$$

We give the following Examples [2.2] to illustrate the function $\Delta_A$.

**Example 2.2.**

(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) If $\mathbb{R}^m$ is endowed with the norm $\|y\|_\infty := \max_{1 \leq i \leq m} |y_i|$ and $A := -\mathbb{R}^m_+$, then $\Delta_A(y) = \max_{1 \leq i \leq m} y_i$.

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

\[
\begin{align*}
B_1 & := \{ y = (y_1, y_2) \in \mathbb{R}^2 : y_1 \leq 0, y_2 > 0 \}, \\
B_2 & := \{ y = (y_1, y_2) \in \mathbb{R}^2 : y_1 - y_2 \leq 0, y_1 > 0 \}, \\
B_3 & := \{ y = (y_1, y_2) \in \mathbb{R}^2 : y_1 - y_2 > 0, y_1 > 0 \}, \\
B_4 & := \{ y = (y_1, y_2) \in \mathbb{R}^2 : y_1 - y_2 > 0, y_1 + y_2 > 0 \}, \\
B_5 & := \{ y = (y_1, y_2) \in \mathbb{R}^2 : y_1 - y_2 \leq 0, y_2 \leq 0 \}.
\end{align*}
\]

Clearly, \( A = B_4 \cup B_5 \) and \( Y \setminus A = B_1 \cup B_2 \cup B_3 \). By a direct calculation, we have

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

and

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

In this paper, for our purposes, let \( A := -C \) in Definition 2.1. For the sake of convenience, we let

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

According to \cite{29} Proposition 3.2 and the fact that \( C \) is a closed, convex and pointed cone with nonempty interior, we have immediately the following properties related to \( \varphi_C \).

**Lemma 2.3.** \cite{29} Let \( \varphi_C(\cdot) \) be defined in \cite{2}. Then the following statements hold:

(i) \( \varphi_C \) is real valued and 1-Lipschitzian;

(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 \( y_1, y_2 \in \mathbb{R}^m \),

\[
\varphi_C(y_1 + y_2) \leq \varphi_C(y_1) + \varphi_C(y_2),
\]

\[
\varphi_C(y_1) - \varphi_C(y_2) \leq \varphi_C(y_1 - y_2);
\]

(vi) Let \( y_1, y_2 \in \mathbb{R}^m \). Then

\[
y_1 < y_2 \Rightarrow \varphi_C(y_1) < \varphi_C(y_2),
\]

\[
y_1 \leq y_2 \Rightarrow \varphi_C(y_1) \leq \varphi_C(y_2).
\]

**Definition 2.4.** \cite{33} For a nonempty set \( M \subset \mathbb{R}^n \), the diameter of \( M \) is defined as

\[
diam(M) := \sup_{x, y \in M} \|x - y\|_2.
\]
Remark 1. If \( M \) is a compact set, then \( \text{diam}(M) = \max_{x,y \in M} \|x - y\|_2 \) and it is a finite number.

Definition 2.5. \[34\] A function \( F : \mathbb{R}^n \to \mathbb{R}^m \) is called \( C \)-convex on \( \mathbb{R}^n \), if for all \( x, y \in \mathbb{R}^n \) and all \( \lambda \in [0, 1] \),

\[
F(\lambda x + (1 - \lambda)y) \leq \lambda F(x) + (1 - \lambda)F(y).
\]

Let \( F : \mathbb{R}^n \to \mathbb{R}^m \) be a vector-valued function with \( F = (f_1, f_2, \ldots, f_m)^\top \), where the superscript \( \top \) denotes the transpose. We say that \( F \) is continuously differentiable if each \( f_i, i \in I := \{1, 2, \ldots, m\} \), is continuously differentiable. Now, let \( F \) be a continuously differentiable function. 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

\[
JF(x) := [\nabla f_1(x) \nabla f_2(x) \ldots \nabla f_m(x)]^\top, \quad x \in \mathbb{R}^n.
\]

Lemma 2.6. \[34\] Assume that the function \( F : \mathbb{R}^n \to \mathbb{R}^m \) is \( C \)-convex on \( \mathbb{R}^n \) and it is continuously differentiable at \( x \in \mathbb{R}^n \). Then

\[
JF(x)(y - x) \preceq F(y) - F(x), \quad \forall y \in \mathbb{R}^n.
\]

3. The vector optimization problem

In this paper, we consider the following constrained vector optimization problems with respect to the partial order \( C \):

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

where \( F = (f_1, f_2, \ldots, f_m)^\top : \Omega \to \mathbb{R}^m \) and \( \Omega \subset \mathbb{R}^n \) is the domain of \( F \) which is assumed to be nonempty, compact and convex. From now on, unless explicitly mentioned, we always assume that \( F \) is continuously differentiable.

Definition 3.1. \[34\] A point \( x \in \Omega \) is called weakly efficient solution of problem \( \text{(3)} \) if there exists no \( x^* \in \Omega \) such that \( F(x^*) \prec F(x) \).

A necessary, but not sufficient, first order optimality condition for problem \( \text{(3)} \) at \( \hat{x} \in \Omega \), is

\[
JF(\hat{x})(\Omega - \hat{x}) \cap (-\text{int}(C)) = \emptyset,
\]

where \( JF(\hat{x})(\Omega - \hat{x}) := \{JF(\hat{x})(s - \hat{x}) : s \in \Omega\} \) and

\[
JF(\hat{x})(s - \hat{x}) = (\nabla f_1(\hat{x}), s - \hat{x}), (\nabla f_2(\hat{x}), s - \hat{x}), \ldots, (\nabla f_m(\hat{x}), s - \hat{x})^\top.
\]
Obviously, (4) is equivalent to $J F(\hat{x})(s - \hat{x}) \notin - \text{int}(C)$ for any $s \in \Omega$.

**Definition 3.2.** A point $\hat{x} \in \Omega$ satisfying (4) is called a stationary point of problem (3).

**Remark 2.** From Lemma 2.3(ii), we can obtain another equivalent characterization of a stationary point $\hat{x}$ of problem (3), i.e.,

$$\varphi_c(J F(\hat{x})(s - \hat{x})) \geq 0, \quad \forall s \in \Omega.$$ 

**Remark 3.** (i) If $m = 1$ and $C := \mathbb{R}_1^+$, then we retrieve the classical stationary condition for constrained scalar optimization problem, i.e., $\langle \nabla f_1(\hat{x}), s - \hat{x} \rangle \geq 0$ for all $s \in \Omega$.
(ii) When $C := \mathbb{R}_m^+$, Definition 3.2 is the same as the notion presented in [11, pp. 744].

**Remark 4.** Note that if $\hat{x} \in \Omega$ is not a stationary point of problem (3), then there exists $\hat{s} \in \Omega$ such that $J F(x)(\hat{s} - \hat{x}) \in - \text{int}(C)$, i.e., $\varphi_c(J F(\hat{x})(\hat{s} - \hat{x})) < 0$ from Lemma 2.3(ii). In this case, as analyzed in [16, pp. 665], we can assert that $\hat{s} - \hat{x}$ is a descent direction for $F$.

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

**Theorem 3.3.** (i) If $\hat{x} \in \Omega$ is a weakly efficient solution of problem (3), 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 (3), then $\hat{x}$ is a weakly efficient solution.

4. Frank-Wolfe method for vector optimization

In this section, we propose an extension of the classical Frank-Wolfe method described in [35, pp. 378] to solve problem (3). First we state and verify that some results that allow us to introduce two types of algorithms. Then, under some additional assumptions, it is proved that all accumulation points of any generated sequences are weakly efficient solution.

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

$$\psi_x(s) := \varphi_c(J F(x)(s - x)), \quad s \in \Omega. \quad (5)$$

**Remark 5.** If we consider the norm $\| \cdot \|_\infty$ in $\mathbb{R}_m^+$ and $C := \mathbb{R}_m^+$, then from Example 2.2(ii), it holds that

$$\psi_x(s) = \max_{i \in I} \langle \nabla f_i(x), s - x \rangle, \quad \forall s \in \Omega.$$ 

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

$$\min_{s \in \Omega} \psi_x(s). \quad (6)$$
Notice that, it follows from Lemma 2.3(iii) that $\psi_x$ defined in (5) is a convex function. This, combined with the fact that $\Omega$ is a nonempty, compact and convex set, gives that problem (6) admits an optimal solution (possibly not unique) on $\Omega$. We denote the optimal solution of problem (6) by $s(x)$, i.e.,

$$s(x) \in \text{argmin}_{s \in \Omega} \psi_x(s).$$

(7)

and the optimal value of problem (6) is denoted by $v(x)$, i.e.,

$$v(x) := \psi_x(s(x)).$$

(8)

According to Remark 4, we formally give the search direction for the objective function $F$ at $x$.

**Definition 4.1.** For any given point $x \in \Omega$, the search direction of the Frank-Wolfe method for $F$ at $x$ is defined as

$$d(x) := s(x) - x,$$

(9)

where $s(x)$ is given by (7).

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 4.2.** Let $v : \Omega \to \mathbb{R}$ be defined in (8). Then, the following statements hold:

(i) $v(x) \leq 0$ for every $x \in \Omega$;
(ii) $x \in \Omega$ is a stationary point of problem (3) if and only if $v(x) = 0$.

**Proof.** (i) Since $x \in \Omega$, it follows from (7) and (8) that $v(x) = \min_{s \in \Omega} \psi_x(s) \leq \psi_x(x) = \varphi_c(JF(x)(x - x)) = \varphi_c(0)$. Besides, $\varphi_c(0) = 0$ by Lemma 2.3(ii). Thus, $v(x) \leq 0$.

(ii) Necessity. Suppose that $x \in \Omega$ is a stationary point of problem (3). Then, it follows from Remark 2 that $\varphi_c(JF(x)(s - x)) \geq 0$ for any $s \in \Omega$. By (7), 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 (8), we obtain $0 = v(x) \leq \psi_x(s) = \varphi_c(JF(x)(s(x) - x))$ for all $s \in \Omega$, which implies that $x$ is a stationary point of problem (3).

**Remark 6.** It is obvious from Proposition 4.2 that $x$ is not a stationary point of problem (3) if and only if $v(x) < 0$.

**Proposition 4.3.** Let $v : \Omega \to \mathbb{R}$ be defined in (8). Then, $v$ is continuous on $\Omega$.

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

$$\limsup_{k \to \infty} v(x^k) \leq v(x) \leq \liminf_{k \to \infty} v(x^k).$$

(10)
Since $s(x) \in \Omega$, using [7] and [8], we can obtain for all $k$,
\[
v(x^k) = \varphi_c(JF(x^k)(s(x^k) - x^k)) \leq \varphi_c(JF(x^k)(s(x) - x^k)).
\]
(11)

Since $F$ is continuously differentiable and $\varphi_c$ is continuous as presented in Lemma 2.3(i), taking $\limsup_{k \to \infty}$ on both sides of inequality in (11), we have
\[
\limsup_{k \to \infty} v(x^k) \leq \varphi_c(JF(x)(s(x) - x)) = v(x).
\]
(12)

Let us show that $v(x) \leq \liminf_{k \to \infty} v(x^k)$. Obviously, we have
\[
v(x) = \min \{ \psi_x(s), s \in \Omega \}
\leq \psi_x(s(x^k))
= \varphi_c(JF(x)(s(x^k) - x))
= \varphi_c(JF(x)(s(x^k) - x^k + x^k - x))
= \varphi_c(JF(x)(s(x^k) - x^k) + JF(x)(x^k - x))
\leq \varphi_c(JF(x)(s(x^k) - x^k)) + \varphi_c(JF(x)(x^k - x)),
\]
where the last inequality follows from Lemma 2.3(v). Taking $\liminf_{k \to \infty}$ in (13), we get
\[
v(x) \leq \liminf_{k \to \infty} \varphi_c(JF(x)(s(x^k) - x^k))
= \liminf_{k \to \infty} (v(x^k) + \varphi_c(JF(x)(s(x^k) - x^k)) - \varphi_c(JF(x^k)(s(x^k) - x^k)))
\leq \liminf_{k \to \infty} (v(x^k) + \|JF(x)(s(x^k) - x^k) - JF(x^k)(s(x^k) - x^k)\|_2)
\leq \liminf_{k \to \infty} (v(x^k) + \|JF(x) - JF(x^k)\|_2\|s(x^k) - x^k\|_2),
\]
(14)
where the penultimate inequality follows from Lemma 2.3(i). Since $s(x^k), x^k \in \Omega$, it follows from Remark 1 that $\|s(x^k) - x^k\| \leq \text{diam}(\Omega) < \infty$. This, combined with the continuously differentiability of $F$ and (14), we get $v(x) \leq \liminf_{k \to \infty} v(x^k)$.

Altogether, (10) holds. Consequently, $v$ is continuous on $\Omega$. \hfill \square

4.1. The Frank-Wolfe method with line search

In this section, we will present the proposed algorithm with line search. In order to compute the stepsize $t > 0$ of our algorithm, we use an Armijo rule. Let $\beta \in (0, 1)$ be a preset constant. The condition to accept $t$ is given by
\[
F(x + t(s(x) - x)) \leq F(x) + t\beta JF(x)(s(x) - x).
\]
(15)

We begin with $t = 1$ and when (15) does not hold, we update
\[
t := \tau t,
\]
where $\tau \in (0, 1)$. The following lemma demonstrates the finiteness of this procedure in view of the fact that (15) holds strictly for $\tau > 0$ small enough.

**Lemma 4.4.** Let $s(x)$ be defined in (7) and $JF(x)(s(x) - x) < 0$. If $\beta \in (0, 1)$, then there exists some $\hat{t} > 0$ such that

$$F(x + t(s(x) - x)) < F(x) + t\beta JF(x)(s(x) - x).$$

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

**Proof.** It is similar to the proof of [13, Proposition 2.1].

Based on the previous discussions, we give the Frank-Wolfe algorithm with Armijio line search (see Algorithm 1) for problem (3). At $k$-th iteration, we compute problem (6) with $x = x_k$. Let us call $s_k := s(x_k)$ and $v_k := v(x_k)$ the optimal solution and optimal value of problem (6) at $k$-th iteration, respectively. The descent direction at $k$-th iteration is computed by $d_k := s_k - x_k$. If $v_k \neq 0$, then we can use $d_k$ with an Armijio line search technique to look for a new solution $x_{k+1}$ which dominates $x_k$.

**Algorithm 1 Frank-Wolfe algorithm with armijo line search**

**Input:** $x^0 \in \Omega$

**for** $k = 0, 1, \ldots$ **do**

- $s^k \leftarrow \arg\min_{s \in \Omega} \psi_{x_k}(s)$
- $v_k \leftarrow \psi_{v_k}(s^k)$
- $d^k \leftarrow s^k - x_k$
- **if** $v_k = 0$ **then**
  - **return** stationary point $x^k$
- **end if**

- $t_k \leftarrow \text{armijo_linear_search}(x_k, d_k, JF(x_k))$
- $x_{k+1} \leftarrow x^k + t_k d^k$

**end for**

Here we describe a vector adaptation for an Armijio line search:

**Algorithm 2 armijo_linear_search**

**Input:** $x^k \in \Omega$, $d^k \in \mathbb{R}^n$, $JF(x^k)$, $\beta \in (0, 1)$, $\tau \in (0, 1)$

$t \leftarrow 1$

**while** $F(x^k + td^k) \not< F(x^k) + t\beta JF(x^k)d^k$ **do**

- $t \leftarrow \tau t$

**end while**

$t_k = t$

Observe that Algorithm 1 ends up with a stationary point in a finite number of iterations or produces an infinite sequence of nonstationary points. From now on, we suppose that Algorithm 1 generates an infinite sequence $\{x^k\}$ of nonstationary points. First, a simple fact that the proposed algorithm generates feasible sequences is given below.

**Theorem 4.5.** Let $\{x^k\}$ be a sequence produced by Algorithm 1. Then, $x^k \in \Omega$ for all $k$. 9
Proof. We proceed by induction. From Algorithm 1, we have \( x^0 \in \Omega \) for \( k = 0 \). Assume that \( x^k \in \Omega \) for \( k > 0 \). We shall prove \( x^{k+1} \in \Omega \) for \( k + 1 \). It is easy to see that \( s^k \in \Omega \) from Algorithm 1. According to the convexity of \( \Omega \), we have \( x^{k+1} = x^k + t_k d^k = x^k + t_k (s^k - x^k) = t_k s^k + (1 - t_k) x^k \in \Omega \) for \( t_k \in (0, 1] \).

We present some properties related to the points which are iterated by Algorithm 1.

Proposition 4.6. For all \( k \), we have

(i) \( v^k < 0 \);
(ii) \( F(x^{k+1}) \prec F(x^k) \);
(iii) \( \sum_{i=0}^k t_i |v^i| \leq \frac{\varphi_c(F(x^i)) - \varphi_c(F(x^{i+1}))}{\beta} \).

Proof. (i) From the assumption that an infinite sequence \( \{x^k\} \) is generated by Algorithm 1 and Remark 6, we have \( v^k < 0 \) for all \( k \).

(ii) By Theorem 4.5, \( x^{k+1} \in \Omega \) for all \( k \). From (15), we have

\[
F(x^{k+1}) \preceq F(x^k) + t_k \beta JF(x^k)(s^k - x^k).
\]

From the nonstationarity of \( x^k \) and Remark 4, we have \( JF(x^k)(s^k - x^k) \prec 0 \). Therefore, the above inequality implies that \( F(x^{k+1}) \prec F(x^k) \).

(iii) For any \( i \), we have

\[
\varphi_c(F(x^{i+1})) \leq \varphi_c(F(x^i)) + t_i \beta JF(x^i)(s^i - x^i)
\]

\[
\leq \varphi_c(F(x^i)) + \varphi_c(t_i \beta JF(x^i)(s^i - x^i))
\]

\[
= \varphi_c(F(x^i)) + t_i \beta \varphi_c(JF(x^i)(s^i - x^i))
\]

\[
= \varphi_c(F(x^i)) + t_i \beta v^i,
\]

where the first inequality holds in view of (15) and Lemma 2.3(vi), the second inequality follows from Lemma 2.3(v), the first equality is due to Lemma 2.3(iv). According to (16) and (i), we have

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

Therefore, adding up from \( i = 0 \) to \( i = k \) in (17), the result is immediately obtained.

Theorem 4.7. Let \( \{x^k\} \) be a sequence produced by Algorithm 1. Then, every accumulation point of \( \{x^k\} \) is a stationary point of problem (3).

Proof. Let \( \hat{x} \in \Omega \) be a accumulation point of the sequence \( \{x^k\} \). Then, there exists a subsequence \( \{x^{k_j}\} \) of \( \{x^k\} \) such that

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

From Proposition 4.3 and (18), 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 4.2(ii).
Let $k := k_j$ in Proposition 4.6(iii). 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,

$$
\lim_{j \to \infty} t_{kj} v(x^{kj}) = 0. \quad (19)
$$

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

(a) $\limsup_{j \to \infty} t_{kj} > 0$ or (b) $\limsup_{j \to \infty} t_{kj} = 0$. \quad (20)

We first suppose that (20)(a) holds. Then, there exists a subsequence $\{t_{kj_i}\}$ of $\{t_{kj}\}$ converging to some $\hat{t} > 0$. And from (18), we have $\lim_{i \to \infty} x_{kj_i} = \hat{x}$. Thus, (19) implies that $\lim_{j \to \infty} t_{kj} v(x^{kj}) = 0$, and furthermore, $\lim_{i \to \infty} v(x^{kj_i}) = 0$. This, combined with Proposition 4.3, gives that $0 = \lim_{i \to \infty} v(x^{kj_i}) = v(\hat{x})$.

We now consider (20)(b). Clearly, $x^{kj_i}, s(x^{kj_i}) \in \Omega$. From the compactness of $\Omega$, Remark 1 and (9), we have $\|d(x^{kj_i})\| = \|s(x^{kj_i}) - x^{kj_i}\| \leq \text{diam}(\Omega) < \infty$, i.e., the sequence $\{d(x^{kj_i})\}$ is bounded. Now, we take subsequences $\{x^{kj_i}\}, \{d(x^{kj_i})\}$ and $\{t_{kj_i}\}$ converging to $\hat{x}$, $d(\hat{x})$ and 0, respectively. By (8) and Proposition 4.6(i), we get

$$
\varphi_c(JF(x^{kj_i})d(x^{kj_i})) = v(x^{kj_i}) < 0.
$$

Taking $\lim_{i \to \infty}$ on both sides of the above inequality and togethering with Proposition 4.3 we have

$$
v(\hat{x}) \leq 0. \quad (21)
$$

Take some fixed but arbitrary $l \in \mathbb{N}$, where $\mathbb{N}$ denotes the set of natural numbers. From $\lim_{i \to \infty} t_{kj_i} = 0$, we have $t_{kj_i} < \tau^l$ for $i$ large enough. This shows that the Armijio condition is not satisfied at $x^{kj_i}$ for $t = \tau^l$, that is,

$$
F(x^{kj_i} + \tau^l d(x^{kj_i})) \not\leq F(x^{kj_i}) + \tau^l \beta JF(x^{kj_i})d(x^{kj_i}),
$$

or, equivalently,

$$
F(x^{kj_i} + \tau^l d(x^{kj_i})) - F(x^{kj_i}) - \tau^l \beta JF(x^{kj_i})d(x^{kj_i}) \not\leq -C,
$$

which means that

$$
F(x^{kj_i} + \tau^l d(x^{kj_i})) - F(x^{kj_i}) - \tau^l \beta JF(x^{kj_i})d(x^{kj_i}) \in \mathbb{R}^m \setminus (-C) = \text{int}(\mathbb{R}^m \setminus (-C)), \quad (22)
$$

11
where the equality holds in view of the closedness of $C$. By (22) and Lemma 2.3(ii), we have

$$\varphi_c(F(x^{k_i} + \tau^l d(x^{k_i})) - F(x^{k_i}) - \tau^l \beta JF(x^{k_i})d(x^{k_i})) > 0. \quad (23)$$

By the continuously differentiability of $F$ and the continuity of $\varphi_c$, taking $\lim_{i \to \infty}$ in (23), we obtain

$$\varphi_c(F(\hat{x} + \tau^l d(\hat{x})) - F(\hat{x}) - \tau^l \beta JF(\hat{x})d(\hat{x})) \geq 0, \quad \forall l \in \mathbb{N}. \quad (24)$$

According to (24), Lemma 4.4 and Lemma 2.3(ii), we can obtain $JF(\hat{x})d(\hat{x}) \not\in 0$, i.e., $JF(\hat{x})d(\hat{x}) \in \mathbb{R}^m \setminus (-\text{int}(C))$. Thus, we have $v(\hat{x}) = \varphi_c(JF(\hat{x})d(\hat{x})) \geq 0$ from Lemma 2.3(ii). This, combined with (21), yields that $v(\hat{x}) = 0$.

It follows from Theorems 3.3 and 4.7 that the following result holds.

**Theorem 4.8.** If $F$ is $C$-convex on $\Omega$, then every accumulation point produced by Algorithm 1 is a weakly efficient solution of problem (3).

### 4.2. The Frank-Wolfe method with adaptive stepsize

In this section, it is always assumed that the gradient of each component of objective function $F$ is Lipschitz continuous with constant $L_i > 0$ for $i \in I$ and we set $L := \max_{i \in I} L_i$. In this sequel, we let

$$e = (e_1, e_2, \ldots, e_m)^\top \in \text{int}(C).$$

We first present an important property which is essential for showing convergence analysis.

**Lemma 4.9.** Suppose that $\frac{L}{2} \| y \|_2^2 e - F(\cdot)$ is $C$-convex on $\Omega$. Then, for any $x, y \in \Omega$,

$$F(y) - F(x) \leq JF(x)(y - x) + \frac{L}{2} \| y - x \|_2^2 e. \quad (25)$$

**Proof.** Let $G(\cdot) := \frac{L}{2} \| \cdot \|_2^2 e - F(\cdot)$. Since $G(\cdot)$ is $C$-convex on $\Omega$, it follows from Lemma 2.6 that

$$JG(x)(y - x) \leq G(y) - G(x). \quad (26)$$

For $JG(x)(y - x)$ in (26), by a simple calculation, we have

$$JG(x)(y - x) = L(x, y)e - L\| x \|_2^2 e - JF(x)(y - x). \quad (27)$$
From the notion of $G(\cdot)$, (26) and (27), we have

$$F(y) - F(x) \preceq JF(x)(y - x) + \frac{L}{2} \|y\|_2^2 e - \frac{L}{2} \|x\|_2^2 e - L(x, y)e + L\|x\|_2^2 e$$

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

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

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

Remark 7. Here we call the property the vector version of the classical descent lemma with respect to the order cone $C$. Actually, the setting of the condition in Lemma 4.9 is inspired by the works of Bauschke et al. [36].

Remark 8. Lemma 4.9 implies that, for any $x, y \in \Omega$,

$$\varphi_c(F(y)) - \varphi_c(F(x)) \leq \psi_x(y(x)) + \frac{L}{2}\|y - x\|_2^2 \varphi_c(e) \tag{28}$$

Indeed, from (25) and Lemma 2.3(iv)–(vi), we have

$$\varphi_c(F(y) - F(x)) \leq \varphi_c \left(JF(x)(y - x) + \frac{L}{2}\|y - x\|_2^2 e\right)$$

$$\leq \varphi_c(JF(x)(y - x)) + \varphi_c \left(\frac{L}{2}\|y - x\|_2^2 e\right) \tag{29}$$

$$= \varphi_c(JF(x)(y - x)) + \frac{L}{2}\|y - x\|_2^2 \varphi_c(e).$$

Obviously, according to Lemma 2.3(vi), it holds that

$$\varphi_c(F(y)) - \varphi_c(F(x)) \leq \varphi_c(F(y) - F(x)). \tag{30}$$

Therefore, it immediately follows from (29), (30) and [3] that (28) holds.

Let us now give the Frank-Wolfe algorithm with adaptive stepsize (see Algorithm [3] for solving problem [3]).

**Algorithm 3** Frank-Wolfe algorithm with adaptive stepsize

**Input:** $x^0 \in \Omega$

**for** $k = 0, 1, \ldots$ **do**

$s^k \leftarrow \arg\min_{s \in \Omega} \psi_{x^k}(s)$

$v^k \leftarrow \psi_{x^k}(s^k)$

$d^k \leftarrow s^k - x^k$

**if** $v^k = 0$ **then**

**return** stationary point $x^k$

**end if**

$t_k \leftarrow \min \left\{1, -\frac{v^k}{L\|d^k\|_2^2} \right\}$

$x^{k+1} \leftarrow x^k + t_k d^k$

**end for**
Likewise, Algorithm 3 can terminate with a stationary point in a finite number of iterations or generate an infinite sequence. We will suppose that in the sequel Algorithm 3 produces an infinite sequence \{x^k\} of nonstationary points. Clearly, it follows from Remark 6 that \( v^k < 0 \) for all \( k \).

**Lemma 4.10.** Suppose that \( \frac{L}{2} \| \cdot \|^2 e - F(\cdot) \) is \( C \)-convex on \( \Omega \), \( \varphi_c(e) < 2 \) and \( \{x^k\} \) is a sequence produced by Algorithm 3. Then, for all \( k \), it holds that

\[
\varphi_c(F(x^{k+1})) - \varphi_c(F(x^k)) \leq \frac{\varphi_c(e) - 2}{2} \min \left\{ \frac{(v^k)^2}{L(\text{diam}(\Omega))^2} - v^k \right\}. \tag{31}
\]

**Proof.** Let \( x^{k+1} = x^k + t_k d^k \), where \( d^k = s^k - x^k \) and

\[
t_k = \min \left\{ 1, -\frac{v^k}{L\|d^k\|^2} \right\}. \tag{32}
\]

Since \( \frac{L}{2} \| \cdot \|^2 e - F(\cdot) \) is \( C \)-convex on \( \Omega \), then by (28) invoked with \( x = x^k \) and \( y = x^{k+1} \), we have

\[
\varphi_c(F(x^{k+1})) - \varphi_c(F(x^k)) \leq \psi_{x^k}(x^k + t_k(s^k - x^k)) + \frac{L}{2} t_k^2 \|d^k\|^2_2 \varphi_c(e)
= t_k \psi_{x^k}(s^k) + \frac{L}{2} t_k^2 \|d^k\|^2_2 \varphi_c(e)
= t_k v^k + \frac{L}{2} t_k^2 \|d^k\|^2_2 \varphi_c(e), \tag{33}
\]

where the first equality holds in view of (5). According to (32), there are two options:

**Case 1.** Let \( t_k = 1 \). This, combined with (32), gives that

\[
L\|d^k\|^2_2 \leq -v^k. \tag{34}
\]

By (33) and (34), we obtain

\[
\varphi_c(F(x^{k+1})) - \varphi_c(F(x^k)) \leq \frac{2 - \varphi_c(e)}{2} v^k. \tag{35}
\]

**Case 2.** Let \( t_k = -\frac{v^k}{L\|d^k\|^2} \). From Remark 1, we get \( \|d^k\| = \|s^k - x^k\| \leq \text{diam}(\Omega) \).

This, together with (33), \( \varphi_c(e) < 2 \), yields that

\[
\varphi_c(F(x^{k+1})) - \varphi_c(F(x^k)) \leq \frac{\varphi_c(e) - 2}{2} \frac{(v^k)^2}{L\|d^k\|^2}
\leq \frac{\varphi_c(e) - 2}{2} \frac{(v^k)^2}{L(\text{diam}(\Omega))^2}. \tag{36}
\]

Therefore, (31) is directly derived by (35) and (36). \( \square \)

To present our convergence analysis for Algorithm 3, we need the following assumption.
**Assumption A.** The sequence \( \{F(x^k)\} \) is \( C \)-bounded from below, i.e., there exists \( \bar{F} \in \mathbb{R}^m \) such that \( \bar{F} \preceq F(x^k) \) for all \( k \).

**Remark 9.** The \( C \)-boundedness is a generalization of the boundedness for scalar value functions. It has been extensively used in the proof of the convergence for gradient-based methods for solving vector optimization problems (see [6,14,15,19]).

**Theorem 4.11.** Suppose that \( \frac{1}{2} \|e - F(\cdot)\|_2^2 \) is \( C \)-convex on \( \Omega \), \( \varphi_c(e) < 2 \) and \( \{x^k\} \) is a sequence produced by Algorithm 3. If Assumption A holds, then every accumulation point of \( \{x^k\} \) is a stationary point of problem (3).

**Proof.** From (31), \( \varphi_c(e) < 2 \) and \( v^k < 0 \), we have for all \( k \),

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

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 \( k \). Since \( \{F(x^k)\} \) is \( C \)-bounded from below (say by \( \bar{F} \)), i.e., \( \bar{F} \preceq F(x^k) \) for all \( k \), it follows from Lemma 2.3(vi) that \( \varphi_c(F) \leq \varphi_c(F(x^k)) \) for all \( k \). Therefore, we know that the sequence \( \{\varphi_c(F(x^k))\} \) is convergent. This obviously means that

\[
\lim_{k \to \infty} \left( \varphi_c(F(x^{k+1})) - \varphi_c(F(x^k)) \right) = 0.
\]

(37)

Taking \( \lim_{k \to \infty} \) in (37), and then combining with (38), we have

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

(39)

From Proposition 4.3, Proposition 4.2(ii) and (39), we obtain that each accumulation point of \( \{x^k\} \) is a stationary point of problem (3).

**Remark 10.** If the \( C \)-convexity of \( F \) is required in the conditions of Theorem 4.11, then it follows from Theorems 3.3 and 4.11 that every accumulation point produced by Algorithm 3 is a weakly efficient solution of problem (3).

**Remark 11.** It is noteworthy that the extended Frank-Wolfe methods for vector optimization problems presented in Algorithms 1 and 3 are conceptual and theoretical schemes rather than implementable algorithms. Similar issues also appear in the literature; see, e.g., [12,13,15,17,20,22,21,27]). Therefore, the computational efficiency of the method to a real-world optimization problem depends essentially on the choice of a good feature and structure of the minimization subproblem (6) at every iteration. For example, when the norm \( \| \cdot \|_\infty \) is used and \( C := \mathbb{R}^m_+ \), the objective function \( \psi_x \) in (6) has the simple form as shown in Remark 5, and then the descent direction can be easily computed by program. We consider this issue as a subject in the following section.
5. Numerical experiments for portfolio optimization

In this section, we present an application of the proposed methods to portfolio optimization problem. The algorithms are were implemented in Python software and ran on a Lenovo computer with Intel(R) Core(TM) i5-8250U processor (1.60 GHz) and 4.0 GB of RAM.

Consider the following bicriteria optimization problem

\[
\min_{\mathbb{R}_+^2} \quad F(x) = \begin{pmatrix} -x^\top u \\ x^\top V x \end{pmatrix}
\]

s.t. \( x \in \Omega \)

where \( u \in \mathbb{R}^n \), \( V \in \mathbb{R}^{n \times n} \) is a symmetric positive semidefinite matrix and

\[
\Omega = \left\{ x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n : \sum_{i=1}^n x_i = 1, x_i \geq 0, i = 1, 2, \ldots, n \right\}.
\]

The problem (40) is actually a well-known portfolio optimization problem, which plays a critical role in determining portfolio strategies for investors. The decision variable \( x = (x_1, x_2, \ldots, x_n) \) of problem (40) stands for the asset weight vector, where \( x_i \), \( i = 1, 2, \ldots, n \), is the weight of asset \( i \) in the portfolio. \( u \) means the return rate of the asset and the variance-covariance matrix \( V = (\sigma_{ij})_{n \times n} \) denotes the variance and covariance of individual asset, where \( \sigma_{ii} \) is the variance of asset \( i \) and \( \sigma_{ij} \) is the covariance between asset \( i \) and asset \( j \). The first objective function denotes the negative of the expected return (that is to be maximized, therefore minimized with a leading minus) and the second one is to minimize the variance of the portfolio, which quantifies the risk associated to the considered portfolio.

Herein, we use the real data presented in [37] that contains five stocks: IBM, Microsoft, Apple, Quest Diagnostics and Bank of America. The expected return and variance of each stock in the portfolio were calculated based on historical stock price and dividend payment from February 1, 2002 to February 1, 2007. Thus, in problem (40), \( n = 5 \),

\[
u = (0.004, 0.00513, 0.04085, 0.01006, 0.01236)
\]

and \( V \) is set as follows:

\[
V = \begin{pmatrix}
0.006461 & 0.002983 & 0.00235487 & 0.00235487 & 0.00096889 \\
0.002983 & 0.0039 & 0.00095937 & -0.0001987 & 0.00063459 \\
0.00235487 & 0.00095937 & 0.01267778 & 0.00135712 & 0.00134481 \\
0.00235487 & -0.0001987 & 0.00135712 & 0.00559836 & 0.00041942 \\
0.00096889 & 0.00063459 & 0.00134481 & 0.00041942 & 0.0016229
\end{pmatrix}.
\]

Considering the operability in practise, we take the norm \( \| \cdot \|_\infty \) in \( \mathbb{R}^2 \) and \( C := \mathbb{R}_{+}^2 \). From Remark 5 and 6, at \( x \in \Omega \), we need solving the following scalar optimization problem

\[
\min_{s \in \Omega} \max_{i=1,2} \langle \nabla f_i(x), s - x \rangle.
\]
Clearly, problem (41) is nondifferentiable. Correspondingly, it can be equivalently transformed into the following differentiable form

\[
\min \gamma \\
\text{s.t. } \gamma \geq \langle \nabla f_i(x), s - x \rangle, i = 1, 2 \\
s \in \Omega
\]  

(42)

Observe that problem (42) is a linear convex optimization problem. Therefore, the optimal solution of problem (42) in our experiment can be obtained by using the \texttt{linprog} of the solver \texttt{optimize} in Python. Moreover, the constrained set is actually an unit simplex. So in order to obtain a set of weakly efficient solutions, we randomly and uniformly sample 50 initial points on the simplex. The stopping criteria in Algorithms 1 and 3 are set as \(|v^k| \leq \epsilon := 10^{-5}\). Algorithms 1 and 3 were respectively run 50 times by using same initial points and each time they ended at solution points, which have been obtained after the verification of the stopping criterion. The solutions obtained by Algorithms 1 and 3 are displayed in Figure 1. Meanwhile, the number of iterations (on the “y” axes) and computing CPU time in seconds (on the “y” axes) for each initial point (50 in total on the “x” axes) are reported in Figure 2(a) and (b), respectively. Note that, in Figure 2(a)–(b), the red and blue dotted lines denote respectively the average of iterations and CPU time obtained by Algorithms 1 and 3 for 50 instances (the specific values are presented in Table 1).

Figure 1. The optimization results of the five stocks.

Table 1. Average of CPU time and the number of iterations.

| Algorithm 1 | Algorithm 3 |
|-------------|-------------|
| Average of CPU time | 0.024 | 0.500 |
| Average of iterations | 7.460 | 169.540 |

Figure 1 shows that some possible optimal portfolio points on a return-risk tradeoff.
As we have seen, the expected return is increasing with the risk. From Figure 2 and Table 1, we observe that Algorithm 1 with Armijo line search takes fewer iterations and CPU time than Algorithm 3 with adaptive stepsize for the same initial points. A reasonable explanation of this phenomenon from the experimental data is that the change of stepsize $t$ in Algorithm 3 for each iteration is very small, which leads to a small improvement of the objective function $F$, so it comes with additional cost.

6. Conclusions

In this paper, we have extended the classical Frank-Wolfe method to solve constrained vector optimization problems with respect to a closed, convex and pointed cone with nonempty interior. A key point is that we construct an auxiliary subproblem via the well-known oriented distance function. Under reasonable assumptions, we prove that accumulation points of the sequences generated by the proposed algorithms with two different strategies of stepsizes are stationary. Applications to portfolio optimization under bicriteria considerations are given.

In recent years, the convergence rate analysis of some gradient-based methods for vector optimization problems have established under the setting of the partial order in $\mathbb{R}^m$ is the nonnegative orthant (see [5],[8],[39]). Moreover, there are some convergence rate results in the case of the general cone order (see [10],[19]). It is noteworthy that in this paper we have not analyzed the convergence rate of the proposed methods. An interesting topic for future research is to investigate this issue.

References

[1] John J. Vector optimization: theory, applications and extensions. 2nd ed. Berlin: Springer; 2011.
[2] Miettinen K. Nonlinear multiobjective optimization. Boston, MAA: Kluwer; 1999.
[3] Eichfelder G. Adaptive scalarization methods in multiobjective optimization. Berlin: Springer; 2008.
[4] Fliege J, Svaiter BF. Steepest descent methods for multicriteria optimization. Math. Methods Oper Res. 2000;51: 479–494.
[5] Fliege J, Graña Drummond LM, Svaiter BF. Newton’s method for multiobjective optimization. SIAM J Optim., 2009;20(2): 602–626.
[6] Qu SJ, Goh M, Chan FTS. Quasi-Newton methods for solving multiobjective optimization. Oper Res Lett. 2011;39(5): 397–399.
[7] Qu SJ, Goh M, Liang B. Trust region methods for solving multiobjective optimisation. Optim Methods Softw. 2013;28(4): 796–811.
[8] Fukuda EH, Graña Drummond L.M.: A survey on multiobjective descent methods. Pesquisa Oper. 2014;34(3): 585–620.
[9] Tanabe H, Fukuda EH, Yamashita N. Proximal gradient methods for multiobjective optimization and their applications. Comput Optim Appl. 2019;72(2): 339–361.
[10] Wang JH, Hu YH, Wai Yu CK, Li C, Yang XQ. Extended Newton methods for multiobjective optimization: majorizing function technique and convergence analysis. SIAM J Optim. 2019;29(3): 339–361.
[11] Assunção PB, Ferreira OP, Prudente LF. Conditional gradient method for multiobjective optimization. Comput Optim Appl. 2021;78(3): 741–768.
[12] Graña Drummond LM, Iusem AN. A projected gradient method for vector optimization problems. Comput Optim Appl. 2004;28(1): 5–29.
[13] Graña Drummond LM, Svaiter BF. A steepest descent method for vector optimization problems. J Comput Appl Math. 2005;175: 395–414.
[14] Fukuda EH, Graña Drummond LM. On the convergence of the projected gradient method for vector optimization. Optimization. 2011;60(8–9): 1009–1021.
[15] Fukuda EH, Graña Drummond LM. Inexact projected gradient method for vector optimization. Comput Optim Appl. 2013;54(3): 473–493.
[16] Graña Drummond LM, Raupp FMP, Svaiter BF. A quadratically convergent Newton method for vector optimization. Optimization. 2014;63(5): 661–677.
[17] Lu F, Chen CR. Newton-like methods for solving vector optimization problems. Appl Anal. 2014;93(8): 1567–1586.
[18] Qu SJ, Goh M, Ji Y, Souza RD. A new algorithm for linearly constrained c-convex vector optimization with a supply chain network risk application. Eur J Oper Res. 2015;247(2): 359–365.
[19] Qu SJ, Ji Y, Jiang JL, Zhang QP. Nonmonotone gradient methods for vector optimization with a portfolio optimization application. Eur J Oper Res. 2017;263(2): 356–366.
[20] Gonçalves MLN, Prudente LF. On the extension of the Hager–Zhang conjugate gradient method for vector optimization. Comput Optim Appl. 2020;76(3): 889–916.
[21] Bonnel H, Iusem AN, Svaiter BF. Proximal methods in vector optimization. SIAM J Optim. 2005;15(4): 953–970.
[22] Ceng LC, Yao JC. Approximate proximal methods in vector optimization. Eur. J Oper Res. 2007;183(1): 1–19.
[23] Chen Z, Huang HQ, Zhao KQ. Approximate generalized proximal-type method for convex vector optimization problem in Banach spaces. Comput Math Appl. 2009;57(7): 1196–1203.
[24] Chuong TD. Tikhonov-type regularization method for efficient solutions in vector optimization. J Comput Appl Math. 2010;234(3): 761–766.
[25] Chuong TD, Yao JC. Steepest descent methods for critical points in vector optimization problems. Appl Anal. 2012;91(10): 1811–1829.
[26] Chuong TD. Newton-like for efficient solutions in vector optimization. Comput Optim Appl. 2013;54: 495–516.
[27] Bot RI, Grad S-M. Inertial forward backward methods for solving vector optimization problems. Optimization. 2018;67(7): 959–974.
[28] Hiriart-Urruty JB. Tangent cone, generalized gradients and mathematical programming in Banach spaces. Math Oper Res. 1979;4: 79–97.
[29] Zaffaroni A. Degrees of efficiency and degrees of minimality. SIAM J Control. Optim.
[30] Miglierina E, Molho E, Rocca M. Well-posedness and scalarization in vector optimization. J Optim Theory Appl. 2005;126(2): 391–409.

[31] Gao Y, Hou SH, Yang XM. Existence and optimality conditions for approximate solutions to vector optimization problems. J Optim Theory Appl. 2012;152(1): 97–120.

[32] Zhou ZA, Chen W, Yang XM. Scalarizations and optimality of constrained set-valued optimization using improvement sets and image space analysis. J Optim Theory Appl. 2019;183: 944–962.

[33] Muscat J. Functional analysis: an introduction to metric spaces, Hilbert Spaces, and Banach Algebras. Springer; 2014.

[34] Luc TD. Theory of vector optimization, lecture notes in economics and mathematical systems, vol. 319, Berlin: Springer; 1989.

[35] Bech A. First-order method in optimization. MPS-SIAM Series on Optimization; SIAM, Philadelphia; 2017.

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

[37] Duan YC. A multi-objective approach to portfolio optimization. Undergraduate Math J. 2007;8(1): 1–18.

[38] Fliege J, Vaz AIF, Vicente LN. Complexity of gradient descent for multiobjective optimization. Optim Methods Soft. 2019;34(5): 949–959.

[39] Tanabe H, Fukuda EH, Yamashita N. Convergence rates analysis of multiobjective proximal gradient methods. arXiv preprint arXiv:2010.08217 2020.