A study of Liu-Storey conjugate gradient methods for vector optimization

M. L. N. Gonçalves[*] F. S. Lima[*] L. F. Prudente[*]

July 2, 2021

Abstract: This work presents a study of Liu-Storey (LS) nonlinear conjugate gradient (CG) methods to solve vector optimization problems. Three variants of the LS-CG method originally designed to solve single-objective problems are extended to the vector setting. The first algorithm restricts the LS conjugate parameter to be nonnegative and use a sufficiently accurate line search satisfying the (vector) standard Wolfe conditions. The second algorithm combines a modification in the LS conjugate parameter with a line search satisfying the (vector) strong Wolfe conditions. The third algorithm consists of a combination of the LS conjugate parameter with a new Armijo-type line search (to be proposed here for the vector setting). Global convergence of the methods under mild assumptions is established. Finally, numerical experiments illustrating the practical efficiency of the new methods and comparisons with existing algorithms are discussed.

Keywords: Vector optimization, Pareto optimality, conjugate gradient methods, global convergence.

AMS subject classifications: 49M15, 65K05, 90C26, 90C29

1 Introduction

Let us first consider the single-objective problem

$$\text{Minimize } f(x), \quad x \in \mathbb{R}^n, \tag{1}$$

where $f\colon \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable. One of the most important methods to solve (1), belonging the class of the nonlinear conjugate gradient (CG) algorithms, is the Liu-Storey (LS) CG method proposed in [36]. Formally, the method generates a sequence $\{x^k\}$ given by

$$x^{k+1} = x^k + \alpha_k d^k, \quad k \geq 0,$$

where the step size $\alpha_k > 0$ is obtained by a line search procedure and the search direction d^k is defined by

$$d^k := \begin{cases} -\nabla f(x^k), & \text{if } k = 0 \\ -\nabla f(x^k) + \beta_k d^{k-1}, & \text{if } k \geq 1, \end{cases} \tag{2}$$

with

$$\beta_k = \beta_k^{LS} := -\frac{\langle \nabla f(x^k), \nabla f(x^k) - \nabla f(x^{k-1}) \rangle}{\langle \nabla f(x^{k-1}), d^{k-1} \rangle}, \tag{3}$$

[*]Instituto de Matemática e Estatística, Universidade Federal de Goiás, CEP 74001-970 - Goiânia, GO, Brazil, E-mails: `maxlng@ufg.br`, `fernandoslima@discente.ufg.br`, `lfprudente@ufg.br`. This work was funded by FAPEG (Grants PRONEM-201710267000532, PPP03/15-201810267001725) and CNPq (Grants 302666/2017-6, 408123/2018-4, 424860/2018-0).

where $\langle \cdot, \cdot \rangle$ denotes the usual inner product. Different choices for β_k in (2) correspond to many existing CG methods. Although too many to name, some notable choices would include: Fletcher-Reeves (FR) [17]; Conjugate Descent (CD) [16]; Dai-Yuan (DY) [13]; Polak-Ribière-Polyak (PRP) [46]; Hestenes-Stiefel (HS) [30].

A distinguish feature of the FR, CD, and DY algorithms is that if a line search satisfying the Wolfe conditions is used, then the corresponding search directions satisfy the so-called *sufficient descent condition*, i.e,

$$\langle \nabla f(x^k), d^k \rangle \leq -c\|\nabla f(x^k)\|^2, \quad \forall k \geq 0,$$

where c is a positive constant and $\|\cdot\|$ denotes the Euclidian norm. However, this is not the case for the PRP, HS, and LS algorithms, that is, these algorithms do not necessarily generate descent directions even when Wolfe line searches are employed. Even more, as shown in [47], the PRP, HS, and LS methods using exact line searches can cycle without approaching a solution of (1). In [23], by assuming that the search directions yield descent, the authors established the convergence of the CG method with $\beta_k := \max\{\beta_k^{PRP}, 0\}$, $\beta_k := \max\{\beta_k^{HS}, 0\}$, and $\beta_k := \max\{\beta_k^{LS}, 0\}$. It is noteworthy that the descent property can be obtained by performing a sufficiently accurate line search; see [23, Section 4] for a careful discussion of this issue. On the other hand, some modifications in the PRP, HS, and LS parameters have been proposed in order to build methods that automatically satisfy the descent property when a usual inexact line search is employed; see, for example, [29, 35, 45]. The algorithms obtained in [29] and [35] use the Wolfe line search conditions and are called Hager-Zhang (HZ-CG) and modified Liu-Storey, respectively. We also mentioned that a LS-CG method with an Armijo-type line search satisfying the sufficient descent condition was proposed in [53].

Consider now the unconstrained vector problem

$$\text{Minimize}_K \; F(x), \quad x \in \mathbb{R}^n, \tag{4}$$

where F is a continuously differentiable function, $K \subset \mathbb{R}^m$ is a closed, convex, and pointed cone with nonempty interior and the partial order in $\mathbb{R}^m$ induced by K (resp. $\text{int}(K)$), $\preceq_K$ (resp. $\prec_K$), is defined by

$$u \preceq_K v \Leftrightarrow v - u \in K \quad (\text{resp. } u \prec_K v \Leftrightarrow v - u \in \text{int}(K)).$$

In the vector context, the concept of optimality is replaced by the concept of *Pareto optimality* or *eficiency.* A point $x^* \in \mathbb{R}^n$ is called Pareto optimal if there is no $x \in \mathbb{R}^n$ with $F(x) \preceq_K F(x^*)$ and $F(x) \neq F(x^*)$. When $K = \mathbb{R}^m_+$, (4) corresponds to the multiobjective optimization problem.

Recently, some conjugate gradient methods of the scalar case were extended to the vector setting. The first work in this line was [40], which extended the FR, CD, DY, PRP, and HS conjugate gradient methods for solving (4). Later, [25] proposed a HZ-CG method for vector optimization problems, whereas [20] studied a CG method for unconstrained quadratic multiobjective problems. Following this research topic, the aim of the present work is to propose and analyze some Liu-Storey conjugate gradient algorithms for solving (4). Basically, we extend the methods considered in [35,36,53] to the vector context. The first algorithm restricts the LS conjugate parameter to be nonnegative and use a line search satisfying the (vector) standard Wolfe conditions. Under the assumption that the directions yield descent, the global convergence of the method is established. We note that the descent property for this particular algorithm can be achieved by performing a sufficiently accurate line search. In the second algorithm, by considering a modification in the LS conjugate parameter, the search directions automatically yield descent regardless of the line search used. The global convergence of the algorithm is provided assuming that the step sizes satisfy the (vector) strong Wolfe conditions. Finally, the third algorithm consists of a combination of the LS parameter with a new Armijo-type line search (to be proposed here for the vector setting). The search directions of the latter algorithm also satisfy the descent condition and its global convergence is proven. From the applicability viewpoint, we test our algorithms on a set of problems taken from the multiobjective optimization literature and comparisons with the steepest descent [19] and PRP [40] methods are also discussed. It is worth mentioning that in the last two decades, the extension of methods of single-objective to vector optimization has been the subject of intense research. Some works on the subject can be found in [1–3, 5–11, 18, 19, 21, 22, 24, 26–28, 38, 49, 55].

This work is organized as follows. Section 2 presents some notation and preliminary results. In section 3, we describe the three proposed LS-CG methods for solving vector optimization problems and present their convergence results, whose proofs are postponed to the Appendix. Numerical results are presented in section 4 and some final remarks are given in section 5.

Notation. If $\mathbb{K} = \{k_1, k_2, \ldots\} \subseteq \mathbb{N}$, with $k_j < k_{j+1}$ for all $j \in \mathbb{N}$, then we denote $\mathbb{K} \underset{\infty}{\subset} \mathbb{N}$. The signum function is denoted by $\operatorname{sgn}(\cdot)$, i.e, given a real number x, then $\operatorname{sgn}(x) = -1, 0$ or 1 if $x < 0$, $x = 0$ or $x > 0$, respectively. The cardinality of a set Ω is denoted by $|\Omega|$.

2 Preliminaries

This sections summarizes some definitions, basic results, and notation of vector optimization used throughout the paper. We refer the reader to [19, 26, 28, 39, 40], where quite useful tools for the vector optimization have been developed.

The positive polar cone of K is the set $K^* := \{w \in \mathbb{R}^m \mid \langle w, y\rangle \geq 0, \forall y \in K\}$. Since K is closed and convex, $K = K^{**}$, $-K = \{y \in \mathbb{R}^m \mid \langle y, w\rangle \leq 0, \forall w \in K^*\}$ and $-\operatorname{int}(K) = \{y \in \mathbb{R}^m \mid \langle y, w\rangle < 0, \forall w \in K^* \setminus \{0\}\}$. Let $C \subset K^* \setminus \{0\}$ be compact and such that

$$K^* = \operatorname{cone}(\operatorname{conv}(C)),$$

i.e., K^* is the conic hull of the convex hull of C. In the multiobjective optimization, we have $K = \mathbb{R}^m_+$, then $K^* = K$ and we can take C as the canonical basis of $\mathbb{R}^m$. If K is a polyhedral cone, K^* is also polyhedral and C can be take as the finite set of extremal rays of K^*. For generic K, the set $C = \{w \in K^* \mid \|w\| = 1\}$ has the desired properties.

The Jacobian of F at x will be denoted by $JF(x)$ and $\operatorname{Image}(JF(x))$ stands for the image on $\mathbb{R}^m$ by the linear operator $JF(x)$. A necessary condition for K-optimality of $x^* \in \mathbb{R}^n$ is

$$-\operatorname{int}(K) \cap \operatorname{Image}(JF(x^*)) = \emptyset.$$

A point $x \in \mathbb{R}^n$ is called *K-critical Pareto* or *stationary* for F when it satisfies the above condition. Therefore, if $x \in \mathbb{R}^n$ is not K-critical, then there exists $v \in \mathbb{R}^n$ such that $JF(x)v \in -\operatorname{int}(K)$ which implies that v is a K-descent direction for F at x, i.e., there exists $\epsilon > 0$ such that $F(x + tv) \prec_K F(x)$ for all $0 < t < \epsilon$; see [39].

Define the function $\varphi\colon \mathbb{R}^m \to \mathbb{R}$ by

$$\varphi(y) := \sup\{\langle y, w\rangle \mid w \in C\}.$$

Since C is compact, φ is well defined. Function φ gives characterizations of $-K$ and $-\operatorname{int}(K)$: $-K = \{y \in \mathbb{R}^m \mid \varphi(y) \leq 0\}$ and $-\operatorname{int}(K) = \{y \in \mathbb{R}^m \mid \varphi(y) < 0\}$.

Lemma 1. *For all y and $y' \in \mathbb{R}^m$, we have*

a) $\varphi(y + y') \leq \varphi(y) + \varphi(y')$ and $\varphi(y) - \varphi(y') \leq \varphi(y - y')$.

b) If $y \preceq_K y'$ then $\varphi(y) \leq \varphi(y')$. If $y \prec_K y'$ then $\varphi(y) < \varphi(y')$.

c) φ is Lipschitz continuous with constant 1.

Proof. See [28, Lemma 3.1]. □

Now define the function $\mathcal{D}\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ by

$$\mathcal{D}(x,d) := \varphi(JF(x)d) = \sup\left\{\langle JF(x)d\,,\, w\rangle \mid w \in C\right\}.$$

Function $\mathcal{D}$ gives a characterization of K-descent directions and of K-critical points, since d is K-descent direction for F at x if $\mathcal{D}(x,d) < 0$ and x is K-critical point for F iff $\mathcal{D}(x,d) \geq 0$ for all d. In connection with the scalar case, we will say that a direction $d \in \mathbb{R}^n$ satisfies the *sufficient descent condition* at $x \in R^n$ iff

$$\mathcal{D}(x,d) \leq c\mathcal{D}(x, d_{SD}(x)), \tag{5}$$

for some $c > 0$. Below, we highlight some other properties of the function $\mathcal{D}$.

Lemma 2. *Let $F\colon \mathbb{R}^n \to \mathbb{R}^m$ be a continuously differentiable function. Then, the following statements holds:*

(a) for $x, y, d \in \mathbb{R}^n$ and $\alpha \geq 0$, we have $\mathcal{D}(x, y + \alpha d) \leq \mathcal{D}(x,y) + \alpha\mathcal{D}(x,d)$;

(b) the mapping $(x,d) \mapsto \mathcal{D}(x,d)$ is continuous;

(c) for $x, y, d \in \mathbb{R}^n$, we have $|\mathcal{D}(x,d) - \mathcal{D}(y,d)| \leq \|JF(x) - JF(y)\|\|d\|$;

(d) if the Jacobian of F is Lipschitz continuous with constant L, then, $\mathcal{D}(\cdot, d)$ is Lipschitz continuous with constant $L\|d\|$.

Proof. Item (a) follows trivially from the definition of $\mathcal{D}(x,d)$ and for the proof of item (b) see [28]. Itens (c) and (d) are direct consequences of the definition of $\mathcal{D}(x,d)$ and Lemma 1(c). □

Define $d_{SD}\colon \mathbb{R}^n \to \mathbb{R}^n$ and $\theta\colon \mathbb{R}^n \to \mathbb{R}$ by

$$d_{SD}(x) := \arg\min\left\{\mathcal{D}(x,d) + \frac{\|d\|^2}{2} \mid d \in \mathbb{R}^n\right\} \tag{6}$$

and

$$\theta(x) := \mathcal{D}(x, d_{SD}(x)) + \frac{\|d_{SD}(x)\|^2}{2}. \tag{7}$$

Since $\mathcal{D}(x,\cdot)$ is a real closed convex function, $d_{SD}(x)$ exists and is unique. The vector $d_{SD}(x)$ extends the notion of the steepest descent direction to the vector optimization setting. Note that in the single-objective case (i.e, $F\colon \mathbb{R}^n \to \mathbb{R}$, $K = \mathbb{R}_+$, and $C = \{1\}$), we have $\mathcal{D}(x,d) = \langle \nabla F(x), d\rangle$, $d_{SD}(x) = -\nabla F(x)$, and $\theta(x) = -\|\nabla F(x)\|^2/2$. For multiobjective optimization, where $K = \mathbb{R}^m_+$ and C is the canonical basis of $\mathbb{R}^m$, $d_{SD}(x)$ can be computed by solving

$$\begin{array}{ll} \text{Minimize} & \alpha + \dfrac{1}{2}\|d\|^2 \\ \text{subject to} & [JF(x)d]_i \leq \alpha, \quad i = 1, \ldots, m, \end{array}$$

which is a convex quadratic problem with linear inequality constraints, see [19]. The following lemma shows that the direction $d_{SD}(x)$ and the optimum value $\theta(x)$ can be used to characterize stationary points of (4).

Lemma 3. *a) If x is K-critical point for F, then $d_{SD}(x) = 0$ and $\theta(x) = 0$.*

b) If x is not K-critical, then $d_{SD}(x) \neq 0$, $\theta(x) < 0$, $\mathcal{D}(x, d_{SD}(x)) < -\|d_{SD}(x)\|^2/2 < 0$ and $d_{SD}(x)$ is K-descent direction for F at x.

c) The mappings $d_{SD}(\cdot)$ and $\theta(\cdot)$ are continuous.

Proof. See [28, Lemma 3.3]. □

Let $d \in \mathbb{R}^n$ be a K-descent direction for F at x and consider the task of finding a step size $\alpha > 0$ from x along d. In connection with the scalar case, it is said that $\alpha > 0$ is obtained by means of an *exact line search* if

$$\mathcal{D}(x + \alpha d, d) = 0.$$

Now, let $e \in K$ be given such that

$$0 < \langle w, e \rangle \leq 1 \text{ for all } w \in C. \tag{8}$$

It is said that $\alpha > 0$ satisfies the *standard Wolfe conditions* if

$$\begin{aligned} F(x + \alpha d) &\preceq_K F(x) + \rho\alpha\mathcal{D}(x, d)e, \\ \mathcal{D}(x + \alpha d, d) &\geq \sigma\mathcal{D}(x, d), \end{aligned}$$

where $0 < \rho < \sigma < 1$ are given constants. In turn, the *strong Wolfe conditions* require $\alpha > 0$ to satisfy

$$\begin{aligned} F(x + \alpha d) &\preceq_K F(x) + \rho\alpha\mathcal{D}(x, d)e, \\ \big|\mathcal{D}(x + \alpha d, d)\big| &\leq \sigma\big|\mathcal{D}(x, d)\big|. \end{aligned}$$

It is easy to see that $e \in K$ as in (8) always exists. In particular, for multiobjective optimization, where $K = \mathbb{R}^m_+$ and C is given by the canonical basis of $\mathbb{R}^m$, we can take $e = [1, \ldots, 1]^T \in \mathbb{R}^m$.

3 Liu-Storey methods for vector optimization

In this section, we propose and analyze three Liu-Storey conjugate gradient (LS-CG) methods for solving vector optimization problems. We first discuss extensions of the classic and modified LS-CG methods considered in [35, 36] and their global convergence results. We then dedicate to a study of a vector version of the LS-CG method with a Armijo-type line search proposed in [53]. From now on, we assume the following hypotheses.

(H) The cone K is polyhedral, C is the (finite) set of normalized extremal rays of K^* and the level set $\mathcal{L} := \{x \mid F(x) \preceq_K F(x^0)\}$ is bounded, where $x^0 \in \mathbb{R}^n$ is the given starting point. In addition, JF is Lipschitz continuous with constant $L > 0$ on an open set containing $\mathcal{L}$, i.e., there exists an open set $\mathcal{N} \supset \mathcal{L}$ such that, for all $x, y \in \mathcal{N}$, it holds that $\|JF(x) - JF(y)\| \leq L\|x - y\|$.

We begin by proposing a generalization of the standard LS-CG method considered in [36]. Here, the LS conjugate parameter is restricted to be nonnegative.

Algorithm 1. Nonnegative Liu-Storey CG method.

Step 0. Let $0 < \rho < \sigma < 1$, $e \in K$ as in (8), and $x^0 \in \mathbb{R}^n$ be given. Initialize $k \leftarrow 0$.

Step 1. Compute $d_{SD}(x^k)$ and $\theta(x^k)$ as in (6) and (7), respectively. If $\theta(x^k) = 0$, then STOP.

Step 2. Define

$$d^k = \begin{cases} d_{SD}(x^k), & \text{if } k = 0 \\ d_{SD}(x^k) + \beta_k d^{k-1}, & \text{if } k \geq 1, \end{cases} \tag{9}$$

where

$$\beta_k := \max\left\{\beta_k^{LS} := \frac{-\mathcal{D}(x^k, d_{SD}(x^k)) + \mathcal{D}(x^{k-1}, d_{SD}(x^k))}{-\mathcal{D}(x^{k-1}, d^{k-1})}, 0\right\} \tag{10}$$

Step 3. Compute $\alpha_k > 0$ such that

$$\begin{aligned} F(x^k + \alpha_k d^k) &\preceq_K F(x^k) + \rho\alpha_k \mathcal{D}(x^k, d^k)e, \\ \mathcal{D}(x^k + \alpha_k d^k, d^k) &\geq \sigma\mathcal{D}(x^k, d^k). \end{aligned}$$

Step 4. Set $x^{k+1} := x^k + \alpha_k d^k$, $k \leftarrow k + 1$, and go to Step 1.

Remark. *(i)* If Algorithm 1 stops at iteration k, then Lemma 3 implies that x^k is K-critical. *(ii)* β_k^{LS} in Step 2 of Algorithm 1 is a direct extension of the classic LS conjugate parameter. Indeed, if $m = 1$, $K = \mathbb{R}_+$, and $C = \{1\}$, we trivially have that β_k^{LS} in (10) corresponds to (3). *(iii)* In Step 3, it is required that the step size α_k satisfies the standard Wolfe conditions. If d^k is a K-descent direction of F at x^k, under assumption **(H)**, it is possible to show that there exist intervals of positive step sizes that satisfy such conditions, see [40, Proposition 3.2]. *(iv)* In the convergence analysis of Algorithm 1, we will require that d^k satisfies the sufficient descent condition (5). As in the scalar case, this descent property is not automatically fulfilled by Algorithm 1, but can be obtained if a sufficiently accurate line search is used. Let us briefly discuss this issue. Since β_k in (10) is non-negative, by the definition of d^k and Lemma 2(a), it follows that

$$\mathcal{D}(x^k, d^k) \leq \mathcal{D}(x^k, d_{SD}(x^k)) + \beta_k \mathcal{D}(x^k, d^{k-1}).$$

Remember that *exact line search* means that $\mathcal{D}(x^k, d^{k-1}) = 0$. Therefore, performing a sufficiently accurate line search it is possible to guarantee that the last term in the above inequality is not too large in order to obtain the desired descent property, see [40,41]. Thus, using inductive arguments, it is possible to show that Algorithm 1 with appropriate line searches is well defined.

We now present the convergence result of Algorithm 1, whose proof is given in Appendix A.

Theorem 1. *Consider Algorithm 1 with a sufficiently accurate line search such that d^k satisfies the sufficient descent condition* (5) *at x^k for every k. Let $\{(x^k, d^k)\}$ be an infinite sequence generated by Algorithm 1. Then,*

$$\liminf_{k \to \infty} \|d_{SD}(x^k)\| = 0.$$

As already mentioned, the Nonnegative LS-CG method does not automatically generates descent directions. In the scalar case, in order to obtain such desired property, a modified LS-CG method was introduced and analyzed in [35]. In the next algorithm, we propose an extension/adaptation of this method for vector optimization, in particular, generating K-descent directions regardless of the line search used. This property is especially interesting in the vector case because it can lead to significant computational savings. Indeed, when a trial step size α is calculated in Step 3 of Algorithm 1 but the associated direction at $x^k + \alpha d^k$ does not satisfy the sufficient descent condition, the computation of $d_{SD}(x^k + \alpha d^k)$ is wasted and the algorithm continues performing a more accurate line search. The modified LS-CG is formally defined as follows.

Algorithm 2. Modified Liu-Storey CG method

Step 0. Let $0 < \rho < \sigma < 1$, $t > 1/2$, $\eta > 0$, $e \in K$ as in (8), and $x^0 \in \mathbb{R}^n$ be given. Initialize $k \leftarrow 0$.

Step 1. Compute $d_{SD}(x^k)$ and $\theta(x^k)$ as in (6) and (7), respectively. If $\theta(x^k) = 0$, then STOP.

Step 2. Define d^k as in (9) with

$$\beta_k := \max\left\{\beta_k^{MLS} := \beta_k^{LS} - t\frac{\|JF(x^k) - JF(x^{k-1})\|^2 \mathcal{D}(x^k, d^{k-1})}{\mathcal{D}^2(x^{k-1}, d^{k-1})}, \eta^k\right\}, \tag{11}$$

where β_k^{LS} is as in (10) and

$$\eta^k := \frac{-1}{\|d^{k-1}\| \min\{\eta, \|d_{SD}(x^{k-1})\|\}}.$$

If $\mathcal{D}(x^k, d^k) > [1 - 1/(2t)]\mathcal{D}(x^k, d_{SD}(x^k))$, then $d^k := d_{SD}(x^k)$ and $\beta_k := 0$.

Step 3. Compute $\alpha_k > 0$ such that

$$F(x^k + \alpha_k d^k) \preceq_K F(x^k) + \rho\alpha_k \mathcal{D}(x^k, d^k)e,$$
$$\left|\mathcal{D}(x^k + \alpha_k d^k, d^k)\right| \leq \sigma\left|\mathcal{D}(x^k, d^k)\right|.$$

Step 4. Set $x^{k+1} := x^k + \alpha_k d^k$, $k \leftarrow k + 1$, and go to Step 1.

We claim that the search direction d^k generated by Algorithm 2 satisfies the sufficient descent condition

$$\mathcal{D}(x^k, d^k) \leq \left(1 - \frac{1}{2t}\right)\mathcal{D}(x^k, d_{SD}(x^k)), \tag{12}$$

for all $k \geq 0$. Assume that this statement holds for d^{k-1} (which is trivially true for d^0). If β_k in (11) is nonnegative, then $\beta_k = \beta_k^{MLS}$ and, by Lemma 2(a), we obtain

$$\begin{aligned}\mathcal{D}(x^k, d^k) &\leq \mathcal{D}(x^k, d_{SD}(x^k)) + \beta_k^{MLS}\mathcal{D}(x^k, d^{k-1}) \\ &\leq \mathcal{D}(x^k, d_{SD}(x^k)) + \frac{\mathcal{D}(x^{k-1}, d_{SD}(x^k)) - \mathcal{D}(x^k, d_{SD}(x^k))}{-\mathcal{D}(x^{k-1}, d^{k-1})}\mathcal{D}(x^k, d^{k-1}) - T_k,\end{aligned} \tag{13}$$

where

$$T_k := t\frac{\Lambda_k^2 \mathcal{D}^2(x^k, d^{k-1})}{\mathcal{D}^2(x^{k-1}, d^{k-1})} \quad \text{and} \quad \Lambda_k := \|JF(x^k) - JF(x^{k-1})\|.$$

It follows from (13) and Lemma 2(c) that

$$\mathcal{D}(x^k, d^k) \leq \mathcal{D}(x^k, d_{SD}(x^k)) + \frac{|\mathcal{D}(x^k, d^{k-1})|\|d_{SD}(x^k)\|\Lambda_k}{-\mathcal{D}(x^{k-1}, d^{k-1})} - T_k.$$

Thus,

$$\mathcal{D}(x^k, d^k) \leq \mathcal{D}(x^k, d_{SD}(x^k)) + \frac{|\mathcal{D}(x^{k-1}, d^{k-1})||\mathcal{D}(x^k, d^{k-1})|\|d_{SD}(x^k)\|\Lambda_k}{\mathcal{D}^2(x^{k-1}, d^{k-1})} - T_k.$$

Taking $a := |\mathcal{D}(x^{k-1}, d^{k-1})|\|d_{SD}(x^k)\|/\sqrt{2t}$ and $b := \sqrt{2t}\Lambda_k \mathcal{D}(x^k, d^{k-1})$, and using that $ab \leq a^2/2 + b^2/2$, we have

$$\mathcal{D}(x^k, d^k) \leq \mathcal{D}(x^k, d_{SD}(x^k)) + \frac{1}{\mathcal{D}^2(x^{k-1}, d^{k-1})}\left[\frac{\|d_{SD}(x^k)\|^2 \mathcal{D}^2(x^{k-1}, d^{k-1})}{4t} + t\Lambda_k^2 \mathcal{D}^2(x^k, d^{k-1})\right] - T_k.$$

Therefore, by the definition of T_k and Lemma 3(b), it follows that

$$\mathcal{D}(x^k, d^k) \leq \mathcal{D}(x^k, d_{SD}(x^k)) + \frac{1}{4t}\|d_{SD}(x^k)\|^2 \leq \left(1 - \frac{1}{2t}\right)\mathcal{D}(x^k, d_{SD}(x^k)),$$

as desired. Now, if β_k in (11) is negative and d^k as in (9) does not satisfy (12), we redefine d^k to be the steepest descent direction, which trivially satisfies the required descent property, proving our claim. The strategy of redefining d^k is supported by the following fact: if β_k in (11) is negative, the descent condition for d^k as in (9) may not hold even when an exact line search is employed. The following example illustrates this case.

Example. *Consider the multiobjective problem* (4) *with:* $K = \mathbb{R}^2_+$*,* C *equal to the canonical basis of* $\mathbb{R}^2$*, and* $F : \mathbb{R}^2 \longrightarrow \mathbb{R}$ *given by*

$$F_1(x) := \frac{x_1^2 + x_2^2}{2} \quad \textit{and} \quad F_2(x) := \frac{(x_1 - 1)^2 - (x_2 - 1)^2}{2}.$$

Define $t = 0.51$*,* $\eta = 0.1$*,* $\sigma = 0.1$*,* $\rho = 10^{-4}$*, and* $e = (1,1)^T$*. Choose* $x^0 = (1,1)^T$*. Solving* (6) *and* (7)*, we obtain* $d_{SD}(x^0) = (-0.6, 0.2)^T$ *and* $\theta(x^0) = 0.2$*. Hence,* $d^0 = (-0.6, 0.2)^T$ *and* $\alpha_0 = 1$ *satisfies the exact line search condition, i.e.,* $\mathcal{D}(x^0 + \alpha_0 d^0, d^0) = 0$*. So,* $x^1 = x^0 + d^0 = (0.4, 1.2)^T$*, for which* $d_{SD}(x^1) = (-34, 10)^T/785$ *and* $\theta(x^1) = 628/785$*. Using* $\mathcal{D}(x^1, d^0) = 0$*, we obtain*

$$\beta_1 = \max\left\{\beta_1^{MLS} = -\frac{18.4}{314}, \eta_1 = -\frac{500}{\sqrt{10}}\right\} = -\frac{18.4}{314},$$

and hence $d^1 := d_{SD}(x^1) + \beta_1 d^0 = (-6.4, 0.8)^T/785$*. Since* $\mathcal{D}(x^1, d^1) = 2.08/785 > 0$*, it follows that* d^1 *is not a K-descent direction for* F *at* x^1*.*

Under Assumption **(H)**, since (12) holds for all k, we obtain from [40, Proposition 3.2] that the line search in Step 3 is well-defined and, as a consequence, Algorithm 2 is also well-defined. Although the strong Wolfe conditions required in Step 3 are not used to show the well-definedness of Algorithm 2, they will be important to ensure convergence. Next theorem presents the convergence result related to Algorithm 2, whose proof is given in Appendix B.

Theorem 2. *Suppose that Algorithm 2 generates an infinite sequence* $\{(x^k, d^k)\}$*. Then,*

$$\liminf_{k\to\infty} \|d_{SD}(x^k)\| = 0.$$

The remaining part of this section is dedicated to propose a vector version of the LS-CG method with an Armijo-type line search proposed in [53]. The potential advantage of this approach is the ease of implementation of the line search, where a simple backtracking procedure can be considered. In [53], in order to get the sufficient descent condition in the scalar minimization, a modification in the classical Armijo condition was proposed in which, for each iteration, the initial trial step size depend on an estimate L_k of the Lipschitz constant L of the gradient of the objective function. We refer the reader to [51, 52] for some estimation techniques for the Lipschitz constant in the scalar case. Now, taking into account the following estimate of the Lipschitz constant L of the Jacobian of F:

$$L \cong \frac{|\mathcal{D}(x^k, d_{SD}(x^k)) - \mathcal{D}(x^{k-1}, d_{SD}(x^k))|}{\|s^{k-1}\|},$$

where $s^{k-1} := x^k - x^{k-1}$, we consider, for each k, the following estimate L_k of L:

$$L_k = \max\left(L_{k-1}, \min\left(\frac{|\mathcal{D}(x^k, d_{SD}(x^k)) - \mathcal{D}(x^{k-1}, d_{SD}(x^k))|}{\|s^{k-1}\|}, \bar{M}\right)\right), \tag{14}$$

where L_0 and $\bar{M}$ are given positive constants such that $L_0 < \bar{M}$. We observe that (14) extends to the vector case a Lipschitz estimate considered in [53].

Algorithm 3. Liu-Storey CG method with Armijo-type line search

Step 0. Let $\rho \in (0,1)$, $\mu \in (0,1)$, $c \in (0,1)$, $L_0 > 0$, $\bar{M} > 0$, $e \in K$ as in (8), and $x^0 \in \mathbb{R}^n$ be given. Compute $d_{SD}(x^0)$ as in (6) and define $d^0 := d_{SD}(x^0)$. Initialize $k \leftarrow 0$.

Step 1. Compute $\theta(x^k)$ as in (7). If $\theta(x^k) = 0$, then STOP.

Step 2. Define $\tau_k := -\frac{(1-c)}{L_k}\frac{\mathcal{D}(x^k,d^k)}{\|d^k\|^2}$, where L_k is as in (14) . Let α_k be the largest value in $\{\tau_k, \tau_k\mu, \tau_k\mu^2, ...\}$ such that

$$F(x^+) \preceq_K F(x^k) + \alpha_k \rho \mathcal{D}(x^k, d^k)e, \tag{15}$$

$$\mathcal{D}(x^+, d(x^+)) \leq c\mathcal{D}(x^+, d_{SD}(x^+)), \tag{16}$$

where

$$x^+ := x^k + \alpha_k d^k, \quad d(x^+) := d_{SD}(x^+) + \left(\frac{-\mathcal{D}(x^+, d_{SD}(x^+)) + \mathcal{D}(x^k, d_{SD}(x^+))}{-\mathcal{D}(x^k, d^k)}\right) d^k.$$

Step 3. Define $x^{k+1} := x^+$, $d_{SD}(x^{k+1}) := d_{SD}(x^+)$, and $d^{k+1} := d(x^+)$. Set $k \leftarrow k+1$, and go to Step 1.

Remark. Note that τ_k is inversely proportional to L_k. Thus, *large* values of L_k can produce *small* step sizes causing slow convergence. This is undesirable behavior of this type of approach even when $L_k = L$ for large values of L. Therefore, in order not to deteriorate the performance of the method, it is important to obtain estimations L_k of L such that they are not so greater than L.

We now present the global convergence result related to Algorithm 3, whose proof is given in Appendix C.

Theorem 3. *Suppose that Algorithm 3 generates an infinite sequence $\{(x^k, d^k)\}$. Then, every limit point of $\{x^k\}$ is a K-critical Pareto point of F.*

4 Numerical experiments

This section presents some numerical results illustrating the practical behavior of the proposed algorithms. Comparisons with the steepest descent method (SD) and the PRP conjugate gradient method proposed in [19] and [40], respectively, are discussed.

We implemented the algorithms in Fortran 90. In order to compute the steepest decent direction $d_{SD}(x)$ and the optimal value $\theta(x)$ in (6) and (7), respectively, we used Algencan [4] which is an augmented Lagrangian code for general nonlinear programming. Regarding the line searches, for computing a step size satisfying the Wolfe conditions, we used the algorithm proposed in [41], which involves several quadratic/cubic polynomial interpolations of the coordinate functions. In turn, the Armijo-type line search considered in Algorithm 3 was implemented as a simple backtracking procedure. We used the following algorithmic parameters: $\rho = 10^{-4}$, $\sigma = 10^{-1}$, $t = 0.75$, $\eta = 10^{-2}$, $c = 10^{-2}$, $\mu = 0.75$, $L_0 = 10^{-4}$, and $\bar{M} = 10^4$. The norm of a matrix $A \in M^{m\times n}$ (see (11)) was computed as

$$\|A\|_{\infty,2} := \max_{x\neq 0} \frac{\|A\|_\infty}{\|x\|} = \max_{i=1,\ldots,m} \|A_{i,\cdot}\| = \max_{i=1,\ldots,m} \left(\sum_{j=1}^n A_{i,j}^2\right)^{1/2}.$$

Since Lemma 3 implies that $x \in R^n$ is a K-critical Pareto point of F iff $\theta(x) = 0$, we stopped the algorithms at x^k reporting convergence when

$$\left|\theta(x^k)\right| \leq 5 \times \texttt{eps}^{1/2},$$

where $\texttt{eps} = 2^{-52} \approx 2.22 \times 10^{-16}$ is the machine precision. The maximum number of allowed iterations was set to 5000. The latter stopping criterion corresponds to failure.

The chosen set of test problems was taken from the literature for multiobjective optimization, i.e., $K = \mathbb{R}^m_+$, C is the canonical basis of $\mathbb{R}^m$, and $e := [1, ..., 1]^T$ in $\mathbb{R}^m$. Table 1 describes the informations for each test problem, where first and second columns contain the name of the problem and the reference where it can be found. Third and fourth columns contain the number of variables and the number of objectives of the corresponding problem, respectively. The starting points were generated belonging to a box $\{x \in \mathbb{R}^n \mid l \leq x \leq u\}$ whose bounds l and u are informed in the last two columns. We emphasize that the boxes reported in Table 1 were used only to define the starting points, but were not considered by the algorithms themselves.

Problem	Source	n	m	l^T	u^T
AP1	[1]	2	3	$(-10, -10)$	$(10, 10)$
AP2	[1]	1	2	-100	100
AP3	[1]	2	2	$(-100, -100)$	$(100, 100)$
AP4	[1]	3	3	$(-10, -10, -10)$	$(10, 10, 10)$
DD1[a]	[14]	5	2	$(-20, \dots, -20)$	$(20, \dots, 20)$
DGO1	[32]	1	2	-10	13
Far1	[32]	2	2	$(-1, -1)$	$(1, 1)$
FDS	[18]	5	3	$(-2, \dots, -2)$	$(2, \dots, 2)$
		50	3	$(-2, \dots, -2)$	$(2, \dots, 2)$
		100	3	$(-2, \dots, -2)$	$(2, \dots, 2)$
		200	3	$(-2, \dots, -2)$	$(2, \dots, 2)$
FF1	[32]	2	2	$(-1, -1)$	$(1, 1)$
Hil1	[31]	2	2	$(0, 0)$	$(1, 1)$
JOS1	[33]	2	2	$(-100, \dots, -100)$	$(100, \dots, 100)$
		50	2	$(-100, \dots, -100)$	$(100, \dots, 100)$
		100	2	$(-100, \dots, -100)$	$(100, \dots, 100)$
		200	2	$(-100, \dots, -100)$	$(100, \dots, 100)$
KW2	[34]	2	2	$(-3, -3)$	$(3, 3)$
Lov1	[37]	2	2	$(-10, -10)$	$(10, 10)$
Lov3	[37]	2	2	$(-20, -20)$	$(20, 20)$
Lov4	[37]	2	2	$(-20, -20)$	$(20, 20)$
Lov5	[37]	3	2	$(-2, -2, -2)$	$(2, 2, 2)$
MGH16[b]	[44]	4	5	$(-25, -5, -5, -1)$	$(25, 5, 5, 1)$
		4	20	$(-25, -5, -5, -1)$	$(25, 5, 5, 1)$
		4	50	$(-25, -5, -5, -1)$	$(25, 5, 5, 1)$
MGH26[b]	[44]	4	4	$(-1, -1, -1 - 1)$	$(1, 1, 1, 1)$
MGH33[b]	[44]	10	10	$(-1, \dots, -1)$	$(1, \dots, 1)$
MLF2	[32]	2	2	$(-100, -100)$	$(100, 100)$
MMR1[c]	[42]	2	2	$(0.1, 0)$	$(1, 1)$
MOP2	[32]	2	2	$(-1, -1)$	$(1, 1)$
MOP3	[32]	2	2	$(-\pi, -\pi)$	(π, π)
MOP5	[32]	2	3	$(-1, -1)$	$(1, 1)$
MOP7	[32]	2	3	$(-400, -400)$	$(400, 400)$
PNR	[48]	2	2	$(-2, -2)$	$(2, 2)$
QV1	[32]	10	2	$(-5, \dots, -5)$	$(5, \dots, 5)$
SK1	[32]	1	2	-100	100
SK2	[32]	4	2	$(-10, -10, -10, -10)$	$(10, 10, 10, 10)$
SLCDT1	[50]	2	2	$(-1.5, -1.5)$	$(1.5, 1.5)$
SLCDT2	[50]	10	3	$(-1, \dots, -1)$	$(1, \dots, 1)$
SP1	[32]	2	2	$(-100, -100)$	$(100, 100)$
SSFYY2	[32]	1	2	-100	100
Toi4[b]	[54]	4	2	$(-2, -2, -2, -2)$	$(5, 5, 5, 5)$
Toi8[b]	[54]	3	3	$(-1, -1, -1, -1)$	$(1, 1, 1, 1)$
Toi9[b]	[54]	4	4	$(-1, -1, -1, -1)$	$(1, 1, 1, 1)$
		50	50	$(-1, -1, -1, -1)$	$(1, 1, 1, 1)$
		100	100	$(-1, -1, -1, -1)$	$(1, 1, 1, 1)$
Toi10[b]	[54]	4	3	$(-2, -2, -2, -2)$	$(2, 2, 2, 2)$
		10	9	$(-2, -2, -2, -2)$	$(2, 2, 2, 2)$
		30	29	$(-2, -2, -2, -2)$	$(2, 2, 2, 2)$
VU1	[32]	2	2	$(-3, -3)$	$(3, 3)$

[a] This is a modified version of DD1 problem that can be found in [18, 43].

[b] This is an adaptation of a single-objective optimization problem to the multiobjective setting that can be found in [43].

[c] This is a modified version of MMR1 problem that can be found in [40].

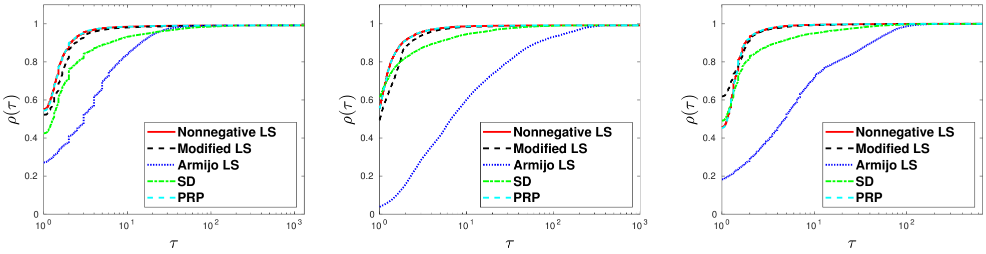
Table 1: List of test problems.

Due to numerical reasons, we considered a scaled version of problem (4) given by

$$\text{Minimize}_K \; \big(\gamma_1 F_1(x), \dots, \gamma_m F_m(x)\big), \quad x \in \mathbb{R}^n, \tag{17}$$

where the scaling factors are computed as $\gamma_j := 1/\max\{1, \|\nabla F_j(x^0)\|_\infty\}$, $j = 1, \dots, m$, where $x^0 \in \mathbb{R}^n$ is the given starting point. It is important to mention that problems (4) and (17) are equivalent, in the sense that they have the same K-critical Pareto points. The numerical comparisons will be shown using performance profiles graphics [15], which we briefly explain here for the sake of completeness. Let $\mathcal{S}$ be the set of solvers, $\mathcal{P}$ be the set of problems, and $t_{p,s} > 0$ be the performance (for example, number of iterations) of the solver $s \in \mathcal{S}$ on the problem $p \in \mathcal{P}$. Here, lower values of $t_{p,s}$ mean better performances. Define the performance ratio $r_{p,s} := t_{p,s}/\min\{t_{p,s} \mid s \in \mathcal{S}\}$ and the cumulative distribution function $\rho_s : [1,\infty) \to [0,1]$ by $\rho_s(\tau) := (1/|\mathcal{P}|)\left|\{p \in \mathcal{P} \mid r_{p,s} \leq \tau\}\right|$. The performance profile is obtained by plotting, for all $s \in \mathcal{S}$, the graph of ρ_s. Note that $\rho_s(1)$ corresponds to the fraction of problems for which solver s was the most efficient over all the considered algorithms. In turn, the *robustness* related to a solver s can be accessed on the extreme right of the graph of ρ_s.

In multiobjective optimization, we are especially interested in estimating the Pareto frontier of a given problem. For this, a strategy is often applied: we run the algorithm at hand from several different starting points and collect the efficient points found. In view of this application, for each problem in Table 1, we considered 300 starting points from a uniform random distribution belonging to the corresponding boxes. Each instance was considered an independent problem and was solved by all the considered solvers. Figure 1 shows the results comparing the algorithms using as the performance measurement: (a) number of iterations; (b) number of functions evaluations; (c) number of steepest descent direction evaluations. In the figure, Algorithms 1, 2, and 3 are identified as Nonnegative LS, Modified LS, and Armijo LS methods, respectively.



(a) Iterations (b) Function evaluations (c) SD evaluations

Figure 1: Performance profiles considering 300 starting points for each test problem using as the performance measurement: (a) number of iterations; (b) number of functions evaluations; (c) number of steepest descent direction evaluations.

As can be seen in Figure 1, the Nonnegative LS and the PRP methods performed almost identically. Regarding the number of iterations, they were the most efficient algorithms followed by the Modified LS, the SD, and the Armijo LS algorithms. With respect to the number of function evaluations, although the SD algorithm was slightly the most efficient, it was quickly outperformed by the CG methods (except the Armijo LS algorithm). Considering the number of steepest descent direction evaluations, the most efficient method was the Modified LS algorithm. This can be explained because the Modified LS algorithm automatically generates descent directions, performing only one steepest descent direction evaluation per iteration. This suggests that the Modified LS algorithm is an attractive alternative for large scale problems for which the computational cost is mainly associated with solving the subproblem (6). The Armijo LS method turned out to be the slowest algorithm, as expected, since it uses a simple line search usually associated with small step sizes. Finally, according to the theoretical results, the methods reached a critical point in all runs.

Next, we compare the ability of the algorithms to properly generate Pareto frontiers. For this, we use the so-called *Purity* and (Γ and Δ) *Spread* metrics. We refer the reader to [12] for a detailed explanation of these metrics. Let $PF_{p,s}$ be an approximation to the Pareto front obtained by solver s for problem p. Here, $PF_{p,s}$ was obtained by running a particular solver s from the same 300 starting points considered in the first part of the experiments and then removing the dominated points (given $x, y \in \mathbb{R}^n$, it is said that x dominates y if $F(y) - F(x) \in \mathbb{R}^m_+ \backslash \{0\}$). Now, let PF_p be an approximation to the Pareto front obtained by first considering $\cup_{s \in \mathcal{S}} PF_{p,s}$ and then removing the dominated points.

- Purity metric: The Purity metric measures the number of nondominated points belonging to PF_p that a solver is able to compute. Given a solver s and a problem p, it is defined by the ratio $\bar{t}_{p,s} := \left|PF_{p,s} \cap PF_p\right| / \left|PF_p\right|$. To analyze the Purity metric using the performance profile, we set $t_{p,s} := 1/\bar{t}_{p,s}$ and, thus, lower values of $t_{p,s}$ indicate better performances. If $\bar{t}_{p,s} = 0$, then we define $t_{p,s} := \infty$. As recommended in [12], we compared the algorithms in pairs when using the Purity metric.

- Spread metrics: A Spread metric seeks to measure the ability of a given solver to obtain *well-distributed* points along the Pareto frontier. Given a solver s and a problem p, consider that $PF_{p,s} \cap PF_p$ is formed by $x_1, \ldots, x_N$ and assume that these points are conveniently sorted by each objective function j such that $F_j(x_i) \leq F_j(x_{i+1})$, $i = 1, ..., N$. Let x_0 and x_{N+1} be the the points corresponding to the lowest and highest values, respectively, of F_j obtained from PF_p. The metrics Γ and Δ are defined by
$$\Gamma_{p,s} := \max_{j \in \{1,\ldots,m\}} \max_{i \in \{0,\ldots,N\}} \delta_{i,j}$$
and
$$\Delta_{p,s} := \max_{j \in \{1,\ldots,m\}} \left(\frac{\delta_{0,j} + \delta_{N,j} + \sum_{i=1}^{N} |\delta_{i,j} - \bar{\delta}_j|}{\delta_{0,j} + \delta_{N,j} + (N-1)\bar{\delta}_j} \right),$$
where $\delta_{i,j} := \left|F_j(x_{i+1}) - F_j(x_i)\right|$ and $\bar{\delta}_j$, $j = 1, \ldots, m$, is the average of the distances $\delta_{i,j}$, $i = 1, ..., N$. In the performance profile, we set $t_{p,s} := \Gamma_{p,s}$ or $t_{p,s} := \Delta_{p,s}$, depending on the chosen metric.

The performance profiles related to the Purity metric are shown in Figure 2 comparing the algorithms in pairs. As we can see, the LS methods slightly outperformed the SD method. On the other hand, no significant difference was noted among the CGs methods.

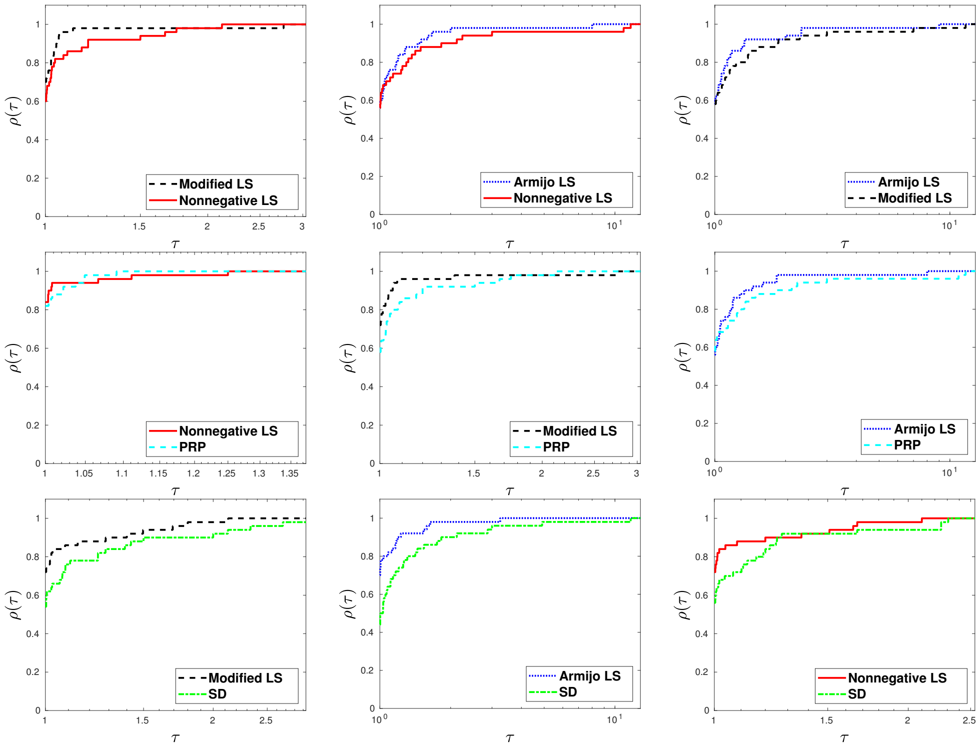
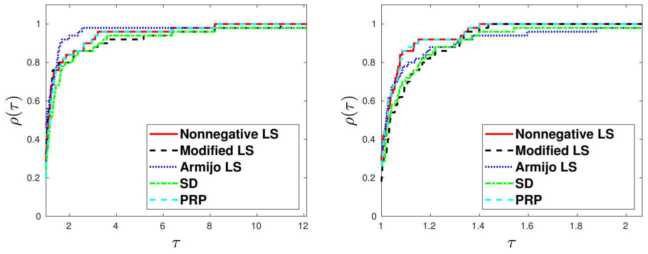


Figure 2: Purity metric performance profiles.

Figure 3 contains the performance profiles of the Γ and Δ Spread metrics. Again, no significant difference among the considered algorithms is noticed for these metrics.



(a) Γ metric (b) Δ metric

Figure 3: Spread metric performance profiles: (a) Γ metric; (b) Δ metric.

5 Final remarks

This paper proposed and analyzed three variants of Liu-Storey (LS) nonlinear conjugate gradient (CG) methods to solve vector optimization problems. These variants are nontrivial extended of LS-CG methods of the scalar case to the vector setting. The global convergences of the new methods were discussed in this general context. Numerical experiments on a set of test problems showed that two of the three methods are competitive with the well-known PRP method in terms of numerical efficiency, as well as to obtain a good representation of Pareto frontiers.

References

[1] M. A. Ansary and G. Panda. A modified Quasi-Newton method for vector optimization problem. *Optimization*, 64(11):2289–2306, 2015.

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

[3] J. Y. Bello Cruz. A subgradient method for vector optimization problems. *SIAM J. Optim.*, 23(4):2169–2182, 2013.

[4] E. Birgin and J. Martinez. *Practical Augmented Lagrangian Methods for Constrained Optimization.* SIAM, Philadelphia, 2014.

[5] H. Bonnel, A. N. Iusem, and B. F. Svaiter. Proximal methods in vector optimization. *SIAM J. Optim.*, 15(4):953–970, 2005.

[6] L. C. Ceng, B. S. Mordukhovich, and J. C. Yao. Hybrid approximate proximal method with auxiliary variational inequality for vector optimization. *J. Optimiz. Theory App.*, 146(2):267–303, 2010.

[7] L. C. Ceng and J. C. Yao. Approximate proximal methods in vector optimization. *Eur. J. Oper. Res.*, 183(1):1–19, 2007.

[8] T. D. Chuong. Generalized proximal method for efficient solutions in vector optimization. *Numer. Funct. Anal. Optim.*, 32(8):843–857, 2011.

[9] T. D. Chuong. Newton-like methods for efficient solutions in vector optimization. *Comput. Optim. Appl.*, 54(3):495–516, 2013.

[10] T. D. Chuong, B. S. Mordukhovich, and J. C. Yao. Hybrid approximate proximal algorithms for efficient solutions in vector optimization. *J. Nonlinear Convex Anal.*, 12(2):257–285, 8 2011.

[11] T. D. Chuong and J. C. Yao. Steepest descent methods for critical points in vector optimization problems. *Appl. Anal.*, 91(10):1811–1829, 2012.

[12] A. L. Custdio, J. F. A. Madeira, A. I. F. Vaz, and L. N. Vicente. Direct multisearch for multiobjective optimization. *SIAM J. Optim.*, 21(3):1109–1140, 2011.

[13] Y. H. Dai and Y. Yuan. A nonlinear conjugate gradient method with a strong global convergence property. *SIAM J. Optim.*, 10(1):177–182, 1999.

[14] I. Das and J. Dennis. Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM J. Optim.*, 8(3):631–657, 1998.

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

[16] R. Fletcher. Practical Method of Optimization, Unconstrained Optimization, vol. 1, 1980.

[17] R. Fletcher and C. M. Reeves. Function minimization by conjugate gradients. *Comput. J.*, 7(2):149–154, 1964.

[18] J. Fliege, L. M. Graña Drummond, and B. F. Svaiter. Newton's method for multiobjective optimization. *SIAM J. Optim.*, 20(2):602–626, 2009.

[19] J. Fliege and B. F. Svaiter. Steepest descent methods for multicriteria optimization. *Math. Method. Oper. Res.*, 51(3):479–494, 2000.

[20] E. H. Fukuda, L. M. G. Drummond, and A. M. Masuda. A conjugate directions-type procedure for quadratic multiobjective optimization. *Optimization*, 0(0):1–19, 2021.

[21] E. H. Fukuda and L. M. Graña Drummond. On the convergence of the projected gradient method for vector optimization. *Optimization*, 60(8-9):1009–1021, 2011.

[22] E. H. Fukuda and L. M. Graña Drummond. Inexact projected gradient method for vector optimization. *Comput. Optim. Appl.*, 54(3):473–493, 2013.

[23] J. C. Gilbert and J. Nocedal. Global convergence properties of conjugate gradient methods for optimization. *SIAM J. Optim.*, 2(1):21–42, 1992.

[24] M. L. N. Gonçalves, F. S. Lima, and L. F. Prudente. Globally convergent Newton-type methods for multiobjective optimization. *http://www.optimization-online.org/DB_FILE/2020/08/7955.pdf*, 2021.

[25] M. L. N. Gonalves and L. F. Prudente. On the extension of the HagerZhang conjugate gradient method for vector optimization. *Comput. Optim. Appl.*, 76(3):889–916, July 2020.

[26] L. M. Graña Drummond and A. N. Iusem. A projected gradient method for vector optimization problems. *Comput. Optim. Appl.*, 28(1):5–29, 2004.

[27] L. M. Graña Drummond, F. M. P. Raupp, and B. F. Svaiter. A quadratically convergent Newton method for vector optimization. *Optimization*, 63(5):661–677, 2014.

[28] L. M. Graña Drummond and B. F. Svaiter. A steepest descent method for vector optimization. *J. Comput. Appl. Math.*, 175(2):395 – 414, 2005.

[29] W. W. Hager and H. Zhang. A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM J. Optim.*, 16(1):170–192, 2005.

[30] M. R. Hestenes and E. Stiefel. *Methods of conjugate gradients for solving linear systems*, volume 49. Journal of Research of the National Bureau of Standards, 1952.

[31] C. Hillermeier. Generalized homotopy approach to multiobjective optimization. *J. Optimiz. Theory App.*, 110(3):557–583, 2001.

[32] S. Huband, P. Hingston, L. Barone, and L. While. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE T. Evolut. Comput.*, 10(5):477–506, 2006.

[33] Y. Jin, M. Olhofer, and B. Sendhoff. Dynamic weighted aggregation for evolutionary multiobjective optimization: Why does it work and how? In *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, GECCO01, page 10421049, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[34] I. Kim and O. de Weck. Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Struct. Multidiscip. O.*, 29(2):149–158, Feb 2005.

[35] M. Li and H. Feng. A sufficient descent LS conjugate gradient method for unconstrained optimization problems. *Appl. Math. Comput.*, 218(5):1577–1586, 2011.

[36] Y. Liu and C. Storey. Efficient generalized conjugate gradient algorithms, part 1: Theory. *J. Optim. Theory Appl.*, 69(1):129137, Apr. 1991.

[37] A. Lovison. Singular continuation: Generating piecewise linear approximations to Pareto sets via global analysis. *SIAM J. Optim.*, 21(2):463–490, 2011.

[38] F. Lu and C.-R. Chen. Newton-like methods for solving vector optimization problems. *Appl. Anal.*, 93(8):1567–1586, 2014.

[39] D. T. Luc. *Theory of vector optimization. Lectures Notes in economics and mathematical systems, vol. 319.* Springer, Berlin, 1989.

[40] L. R. Lucambio Pérez and L. F. Prudente. Nonlinear conjugate gradient methods for vector optimization. *SIAM J. Optim.*, 28(3):2690–2720, 2018.

[41] L. R. Lucambio Pérez and L. F. Prudente. A Wolfe line search algorithm for vector optimization. *ACM Trans. Math. Softw.*, 45(4), Dec. 2019.

[42] E. Miglierina, E. Molho, and M. Recchioni. Box-constrained multi-objective optimization: A gradient-like method without a priori scalarization. *Eur. J. Oper. Res.*, 188(3):662–682, 2008.

[43] K. Mita, E. H. Fukuda, and N. Yamashita. Nonmonotone line searches for unconstrained multiobjective optimization problems. *J. Global Optim.*, 75:63–90, 2019.

[44] J. J. Moré, B. S. Garbow, and K. E. Hillstrom. Testing unconstrained optimization software. *ACM Trans. Math. Softw.*, 7(1):17–41, Mar. 1981.

[45] Y. Narushima and H. Yabe. A survey of sufficient descent conjugate gradient methods for unconstrained optimization. *SUT J. Math.*, 50(2):167–203, 2014.

[46] E. Polak and G. Ribière. Note sur la convergence de méthodes de directions conjuguées. *Revue française d'informatique et de recherche opérationnelle, série rouge*, 3(1):35–43, 1969.

[47] M. J. D. Powell. Nonconvex minimization calculations and the conjugate gradient method. In *Numerical analysis*, pages 122–141. Springer, 1984.

[48] M. Preuss, B. Naujoks, and G. Rudolph. Pareto set and EMOA behavior for simple multimodal multiobjective functions. In T. P. Runarsson, H.-G. Beyer, E. Burke, J. J. Merelo-Guervós, L. D. Whitley, and X. Yao, editors, *Parallel Problem Solving from Nature - PPSN IX*, pages 513–522, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[49] S. Qu, M. Goh, and F. T. Chan. Quasi-Newton methods for solving multiobjective optimization. *Oper. Res. Lett.*, 39(5):397–399, 2011.

[50] O. Schütze, M. Laumanns, C. A. Coello Coello, M. Dellnitz, and E.-G. Talbi. Convergence of stochastic search algorithms to finite size Pareto set approximations. *J. Global Optim.*, 41(4):559–577, Aug 2008.

[51] Z.-J. Shi and J. Shen. Convergence of descent method without line search. *Appl. Math. Comput.*, 167(1):94–107, 2005.

[52] Z.-J. Shi and J. Shen. Step-size estimation for unconstrained optimization methods. *Comput. Appl. Math.*, 24(3):399–416, 2005.

[53] Z.-J. Shi and J. Shen. Convergence of LiuStorey conjugate gradient method. *Eur. J. Oper. Res.*, 182(2):552–560, 2007.

[54] P. L. Toint. Test problems for partially separable optimization and results for the routine PSPMIN. *The University of Namur, Department of Mathematics, Belgium, Tech. Rep.*, 1983.

[55] K. D. Villacorta and P. R. Oliveira. An interior proximal method in vector optimization. *Eur. J. Oper. Res.*, 214(3):485–492, 2011.

Appendix

Assume that $\{x^k\}$ is a sequence generated by one of the proposed algorithms. Since $F(x^k) \preceq_K F(x^0)$ for all k, we obtain $\{x^k\} \subset \mathcal{L}$. Thus, taking into account assumption **(H)**, it follows that $\{x^k\}$ is bounded. So, there exists $\bar{C} > 0$ such that $\|x^k\| \leq \bar{C}$ for all k. Thus, it follows from Lemma 3(c) and the continuity of JF that there exist positive constants $\bar{\gamma} > 0$ and $\bar{c} > 0$ such that

$$\|d_{SD}(x^k)\| \leq \bar{\gamma}, \quad \|JF(x^k)\| \leq \bar{c}, \quad \forall k \geq 0. \tag{18}$$

Moreover, for all $w \in C$ and $k \geq 0$, it turns out that

$$0 > \mathcal{D}(x^k, d_{SD}(x^k)) \geq \langle JF(x^k)d_{SD}(x^k), w\rangle \geq -\|JF(x^k)\|\|d_{SD}(x^k)\| \geq -\bar{c}\bar{\gamma}, \tag{19}$$

because $\|w\| = 1$.

A Proof of Theorem 1

In the single-objective case, the global convergence of the (non-negative) PRP, HS and LS conjugate gradient methods were obtained by means of the so-called Property $(*)$ introduced in [23]; see also [45]. This property was extended to the vector setting in [40]. We will use Property $(*)$, which is formally stated below, to prove the convergence of Algorithm 1.

Property (∗). *Consider a CG method and suppose that there exist $\gamma > 0$ such that*

$$0 < \gamma \leq \|d_{SD}(x^k)\|, \quad \forall k \geq 0. \tag{20}$$

Under this assumption, we say that the method has Property (∗) if there exist constants $b > 1$ and $\lambda > 0$ such that for all k:

$$|\beta_k| \leq b,$$

and

$$\|s^{k-1}\| \leq \lambda \Rightarrow |\beta_k| \leq \frac{1}{2b},$$

where $s^{k-1} = x^k - x^{k-1}$.

The next result shows that, under mild assumptions, a (vector) CG method that has Property (∗) is globally convergent. Since its proof is similar to [40, Theorem 5.10], it is omitted here.

Theorem 4. *Consider a CG method with the following characteristics: (i) $\beta_k \geq 0$ for all k; (ii) d^k satisfies the sufficient descent condition (5) at x^k for all k; (iii) α_k satisfies the standard Wolfe conditions for all k; (iv) Property (∗) holds. Then,*

$$\liminf_{k \to \infty} \|d_{SD}(x^k)\| = 0.$$

We are now ready to prove Theorem 1.

Proof of Theorem 1: In view of Theorem 4, it suffices to show that Algorithm 1 has Property (∗). As observed in [45, pp.174], in order to show that a CG method has Property (∗), it is suffices to prove that, under assumption (20), there is a positive constant M such that

$$|\beta_k| \leq M\|s^{k-1}\|, \quad \forall k \geq 0. \tag{21}$$

Assume that (20) holds. Then, by (18) and (20), we have

$$0 < \gamma \leq \|d_{SD}(x^k)\| \leq \bar{\gamma}, \quad \forall k \geq 0. \tag{22}$$

Moreover, by Lemma 3(b) and (19), it follows that

$$\frac{\gamma^2}{2} \leq -\mathcal{D}(x^k, d_{SD}(x^k)) \leq \bar{c}\bar{\gamma}. \tag{23}$$

By **(H)**, Lemma 2(d) and (22), we obtain

$$\left| -\mathcal{D}(x^k, d_{SD}(x^k)) + \mathcal{D}(x^{k-1}, d_{SD}(x^k)) \right| \leq L\|d_{SD}(x^k)\|\|x^k - x^{k-1}\| = L\bar{\gamma}\|s^{k-1}\|. \tag{24}$$

Since (x^k, d^k) satisfies the sufficient descent condition (5), it follows from (24) and the first inequality in (23) that

$$\left|\beta_k^{LS}\right| \leq \frac{|-\mathcal{D}(x^k, d_{SD}(x^k)) + \mathcal{D}(x^{k-1}, d_{SD}(x^k))|}{-\mathcal{D}(x^{k-1}, d^{k-1})} \leq \frac{L\bar{\gamma}\|s^{k-1}\|}{-c\mathcal{D}(x^{k-1}, d_{SD}(x^{k-1}))} \leq \frac{2L\bar{\gamma}}{c\gamma^2}\|s^{k-1}\|.$$

By defining $M = 2L\bar{\gamma}/c\gamma^2$, we conclude that (21) holds. Therefore, Algorithm 1 has Property (∗), completing the proof. □

B Proof of Theorem 2

Let us first establish some estimates on the parameter β_k of Algorithm 2 which will help us in the convergence analysis.

Lemma 4. *Consider that Algorithm 2 generates an infinite sequence* $\{(x^k, d^k)\}$. *Assume also that* (20) *holds. Then, there exists* $M > 0$ *such that*

$$\eta^k \leq \beta_k, \qquad |\beta_k| \leq M\|s^{k-1}\|, \quad \forall k \geq 0.$$

Proof. The first inequality follows from the definition of β_k in (11). Now, as in Appendix A, we have that (18), (19), (22), and (23) hold for all $k \geq 0$. In particular, we obtain

$$\Lambda_k := \|JF(x^k) - JF(x^{k-1})\| \leq 2\bar{c}.$$

Therefore, from the definition of β_k, Lemma 2(d), the last inequality, and the fact that JF is Lipschitz continuous, we have

$$\begin{aligned}|\beta_k| \leq |\beta_k^{MLS}| &\leq \frac{|\mathcal{D}(x^{k-1}, d_{SD}(x^k)) - \mathcal{D}(x^k, d_{SD}(x^k)|}{-\mathcal{D}(x^{k-1}, d^{k-1})} + t\Lambda_k^2 \frac{|\mathcal{D}(x^k, d^{k-1})|}{\mathcal{D}^2(x^{k-1}, d^{k-1})} \\ &\leq \frac{L\|d_{SD}(x^k)\|\|s^{k-1}\|}{-\mathcal{D}(x^{k-1}, d^{k-1})} + t\frac{2\bar{c}L\|s^{k-1}\||\mathcal{D}(x^k, d^{k-1})|}{\mathcal{D}^2(x^{k-1}, d^{k-1})}.\end{aligned}$$

Thus, using the strong Wolfe conditions at Step 3 of Algorithm 2 and the sufficient descent conditions (12) with $c := 1 - 1/(2t)$, it follows that

$$\begin{aligned}|\beta_k| &\leq \frac{L\|d_{SD}(x^k)\|\|s^{k-1}\|}{-\mathcal{D}(x^{k-1}, d^{k-1})} + t\frac{2\bar{c}L\|s^{k-1}\|\sigma}{-\mathcal{D}(x^{k-1}, d^{k-1})} \\ &\leq \frac{L\|d_{SD}(x^k)\|\|s^{k-1}\|}{-c\mathcal{D}(x^{k-1}, d_{SD}(x^{k-1}))} + t\frac{2\bar{c}L\|s^{k-1}\|\sigma}{-c\mathcal{D}(x^{k-1}, d_{SD}(x^{k-1}))}.\end{aligned}$$

Hence, from (22) and (23), we obtain

$$|\beta_k| \leq \|s^{k-1}\| \left(2\frac{L\bar{\gamma} + 2t\bar{c}\sigma L}{c\gamma^2}\right).$$

By defining $M := 2(L\bar{\gamma} + 2t\bar{c}\sigma L)/c\gamma^2$, we obtain the second inequality of the lemma. □

The next result will be useful to establish the convergence of Algorithm 2.

Lemma 5. *Consider that Algorithm 2 generates an infinite sequence* $\{(x^k, d^k)\}$. *Assume that* (20) *holds. Then,* $d^k \neq 0$ *for all* $k \geq 0$, *and*

$$\sum_{k=1}^{\infty} \frac{1}{\|d^k\|^2} < \infty, \qquad \sum_{k=1}^{\infty} \|u^k - u^{k-1}\|^2 < \infty, \quad \textit{where} \quad u^k := \frac{d^k}{\|d^k\|}.$$

Proof: Using Lemma 4, the proof follows similarly to the one presented in [25, Lemma 6] . □

Proof of Theorem 2: Using Lemmas 4 and 5, the proof follows by contradiction similarly to the one presented in [25, Theorem 2]. □

C Proof of Theorem 3

In order to prove Theorem 3, we first present some technical results.

Lemma 6. *Assume that $d \in \mathbb{R}^n$ is a K-descent direction for F at x. Let $\rho \in (0,1)$, $c \in (0,1)$, and $e \in K$ as in (8) be given. Then, there exists $\bar{\alpha} > 0$ such that, for all $\alpha \in (0, \bar{\alpha})$,*

$$F(x + \alpha d) \preceq_K F(x) + \alpha \rho \mathcal{D}(x, d) e, \tag{25}$$

$$\mathcal{D}(x + \alpha d, d(x + \alpha d)) \leq c \mathcal{D}(x + \alpha d, d_{SD}(x + \alpha d)), \tag{26}$$

where

$$d(x + \alpha d) := d_{SD}(x + \alpha d) + \left(\frac{-\mathcal{D}(x + \alpha d, d_{SD}(x + \alpha d)) + \mathcal{D}(x, d_{SD}(x + \alpha d))}{-\mathcal{D}(x, d)} \right) d.$$

Proof. Since $\mathcal{D}(x, d) < 0$, it is easy to see that there exists $\alpha' > 0$ such that (25) holds for all $\alpha \in (0, \alpha')$; see, for example, [41, Lemma 1]. Now, define $\phi, \psi : \mathbb{R}_+ \longrightarrow \mathbb{R}$ by

$$\phi(\alpha) := \mathcal{D}(x + \alpha d, d(x + \alpha d)) \quad \text{and} \quad \psi(\alpha) := c\mathcal{D}(x + \alpha d, d_{SD}(x + \alpha d)).$$

Note that $\phi(0) < \psi(0)$. Thus, by continuity arguments, there exists $\alpha'' > 0$ such that $\phi(\alpha) \leq \psi(\alpha)$ for all $\alpha \in (0, \alpha'')$, i.e., (26) holds for all $\alpha \in (0, \alpha'')$. Therefore, by taking $\bar{\alpha} := \min(\alpha', \alpha'')$, we conclude that (25) and (26) hold for all $\alpha \in (0, \bar{\alpha})$. □

Lemma 7. *Algorithm 3 is well-defined.*

Proof. By using induction on k, the proof follows straightforwardly from Lemma 6. □

Lemma 8. *There holds:*

$$\|d^k\| \leq \left(1 + \frac{L(1-c)}{L_0} \right) \|d_{SD}(x^k)\|, \quad \forall k \geq 0. \tag{27}$$

Proof. For $k = 0$, we have $d^0 = d_{SD}(x^0)$ and hence (27) is trivially satisfied. For $k \geq 1$, since $L_{k-1} \geq L_0$, it follows from the definition of α_{k-1} in Step 2 of Algorithm 3 that

$$\alpha_{k-1} \leq -\frac{(1-c)}{L_{k-1}} \frac{\mathcal{D}(x^{k-1}, d^{k-1})}{\|d^{k-1}\|^2} \leq -\frac{(1-c)}{L_0} \frac{\mathcal{D}(x^{k-1}, d^{k-1})}{\|d^{k-1}\|^2}, \tag{28}$$

because $\mathcal{D}(x^{k-1}, d^{k-1}) < 0$. Now, by the definition of d^k, assumption **(H)**, Lemma 2(d), and the fact that $x^k - x^{k-1} = \alpha_{k-1} d^{k-1}$, we have

$$\begin{aligned} \|d^k\| &\leq \|d_{SD}(x^k)\| + \frac{|-\mathcal{D}(x^k, d_{SD}(x^k)) + \mathcal{D}(x^{k-1}, d_{SD}(x^k))|}{-\mathcal{D}(x^{k-1}, d^{k-1})} \|d^{k-1}\| \\ &\leq \|d_{SD}(x^k)\| + \frac{L\|d_{SD}(x^k)\|\|x^k - x^{k-1}\|}{-\mathcal{D}(x^{k-1}, d^{k-1})} \|d^{k-1}\| \\ &= \|d_{SD}(x^k)\| \left(1 + \frac{L\alpha_{k-1}\|d^{k-1}\|^2}{-\mathcal{D}(x^{k-1}, d^{k-1})} \right), \end{aligned}$$

which, combined with (28), yields the desired inequality (27). □

Lemma 9. *There exists $\tau_{\min} > 0$ such that $\tau_k > \tau_{\min}$ for all $k \geq 0$.*

Proof. First, note that $L_k \leq \bar{M}$ for all $k \geq 0$. Thus, from the definition of τ_k, (16), and Lemma 3(b), we have

$$\tau_k = -\frac{(1-c)}{L_k}\frac{\mathcal{D}(x^k, d^k)}{\|d^k\|^2} \geq -\frac{c(1-c)}{\bar{M}}\frac{\mathcal{D}(x^k, d_{SD}(x^k))}{\|d^k\|^2} > \frac{c(1-c)}{2\bar{M}}\frac{\|d_{SD}(x^k)\|^2}{\|d^k\|^2}.$$

Then, using (27), it follows that

$$\tau_k > \frac{c(1-c)}{2\bar{M}}\left(1 + \frac{L(1-c)}{L_0}\right)^{-2}.$$

By defining $\tau_{\min}$ as the right hand side of the above inequality, we conclude the proof. □

We are now ready to prove Theorem 3.

Proof of Theorem 3: Let x^* be a limit point of the sequence $\{x^k\}$. Thus, there exists $\mathbb{K} := \{k_0, k_1, k_2, ...\} \underset{\infty}{\subset} \mathbb{N}$ such that $\lim_{k \in K} x^k = x^*$. Hence, by continuity of F, we have

$$\lim_{k \in \mathbb{K}} F(x^k) = F(x^*).$$

Noting that (8) implies $\varphi(-e) < 0$, by (15) and Lemma 1, we obtain, for all $k \geq 0$,

$$\varphi(F(x^{k+1})) \leq \varphi(F(x^k)) + \rho\alpha_k\varphi(\mathcal{D}(x^k, d^k)e) = \varphi(F(x^k)) - \mathcal{D}(x^k, d^k)\rho\alpha_k\varphi(-e) \leq \varphi(F(x^k)),$$

which, in particular, implies that $\{\varphi(F(x^k))\}$ is a nonincreasing sequence. Therefore, since $k_{j+1} \geq k_j + 1$, we have

$$\varphi(F(x^{k_{j+1}})) \leq \varphi(F(x^{k_j+1})) \leq \varphi(F(x^{k_j})) - \mathcal{D}(x^{k_j}, d^{k_j})\rho\alpha_{k_j}\varphi(-e) \leq \varphi(F(x^{k_j})),$$

for all $j \geq 0$. Then, by using continuity arguments, we have

$$\lim_{j \to \infty} \alpha_{k_j}\mathcal{D}(x^{k_j}, d^{k_j}) = 0.$$

As a consequence, there exists $\mathbb{K}_1 \underset{\infty}{\subset} \mathbb{K}$ such that at least one of the two following possibilities is fulfilled: (a) $\lim_{k \in \mathbb{K}_1} \mathcal{D}(x^k, d^k) = 0$; (b) $\lim_{k \in \mathbb{K}_1} \alpha_k = 0$.
Case (a): By (16) and Lemma 3(b), it follows that

$$\mathcal{D}(x^k, d^k) \leq c\mathcal{D}(x^k, d_{SD}(x^k)) \leq -\frac{c}{2}\|d_{SD}(x^k)\|^2 \leq 0.$$

Thus, taking limits for $k \in \mathbb{K}_1$, we conclude that $\lim_{k \in \mathbb{K}_1} d_{SD}(x^k) = 0$. Then, $d_{SD}(x^*) = 0$ and, hence, x^* is K-critical Pareto point of F.
Case (b): Without loss of generality, assume that $\alpha_k < \tau_{\min}$ for all $k \in \mathbb{K}_1$, where $\tau_{\min} > 0$ is given as in Lemma 9. Thus, by Step 2 of the algorithm and Lemma 9, defining $\bar{\alpha}_k := \alpha_k/\mu$, we conclude that, for each $k \in \mathbb{K}_1$, at least one of the two following possibilities holds:

$$F(\bar{x}^k) \succ_K F(x^k) + \bar{\alpha}_k\rho\mathcal{D}(x^k, d^k)e, \tag{29}$$

or

$$\mathcal{D}(\bar{x}^k, d(\bar{x}^k)) > c\mathcal{D}(\bar{x}^k, d_{SD}(\bar{x}^k)), \tag{30}$$

where $\bar{x}^k := x^k + \bar{\alpha}_k d^k$. Assume that (29) holds infinitely many times, say for $\mathbb{K}_2 \underset{\infty}{\subset} \mathbb{K}_1$. Then, for all $k \in \mathbb{K}_2$, there exists $w_k \in C$ such that

$$\langle F(\bar{x}^k) - F(x^k), w_k\rangle > \bar{\alpha}_k\rho\mathcal{D}(x^k, d^k)\langle e, w_k\rangle \geq \bar{\alpha}_k\rho\mathcal{D}(x^k, d^k), \tag{31}$$

where the last inequality follows from (8). By the mean value theorem, there exists $\epsilon_k \in (0,1)$ such that

$$F(\bar{x}^k) - F(x^k) = \bar{\alpha}_k JF(x^k + \epsilon_k \bar{\alpha}_k d^k) d^k. \tag{32}$$

So, combining (31) and (32), and using the definition of $\mathcal{D}(\cdot,\cdot)$, we obtain

$$\rho \mathcal{D}(x^k, d^k) < \langle JF(x^k + \epsilon_k \bar{\alpha}_k d^k) d^k, w_k \rangle \leq \mathcal{D}(x^k + \epsilon_k \bar{\alpha}_k d^k, d^k), \tag{33}$$

for all $k \in \mathbb{K}_2$. It follows from Lemma 8 and (18) that $\{d^k\}$ is bounded and, as a consequence, there exist $\mathbb{K}_3 \underset{\infty}{\subset} \mathbb{K}_2$ and $d^* \in \mathbb{R}^n$ such that $\lim_{k \in \mathbb{K}_3} d^k = d^*$. Moreover, by the definition of $\bar{\alpha}_k$, we obtain $\lim_{k \in \mathbb{K}_3} \bar{\alpha}_k = 0$. Therefore, taking limits in (33) for $k \in \mathbb{K}_3$ and using Lemma 2(b), we get

$$\rho \mathcal{D}(x^*, d^*) \leq \mathcal{D}(x^*, d^*).$$

Since $\rho < 1$ and $\mathcal{D}(x^k, d^k) < 0$ for all k, this implies that $\mathcal{D}(x^*, d^*) = 0$. On the other hand, by (16) and Lemma 3(b), we have

$$\mathcal{D}(x^k, d^k) \leq c\mathcal{D}(x^k, d_{SD}(x^k)) \leq -\frac{c}{2}\|d_{SD}(x^k)\|^2 \leq 0.$$

Therefore, taking limits in the above inequality for $k \in \mathbb{K}_3$, we conclude that $d_{SD}(x^*) = 0$, proving that x^* is a K-critical Pareto point of F.

Assume now, without loss of generality, that (30) holds and (29) is not true for all $k \in \mathbb{K}_1$. Thus, $\{\bar{x}^k\}_{k \in \mathbb{K}_1} \subset \mathcal{L}$, which implies, by using continuity arguments, that there exists $\bar{c} > 0$ such that

$$\|JF(\bar{x}^k)\| \leq \bar{c}, \quad \forall k \in \mathbb{K}_1. \tag{34}$$

Define $\chi := \operatorname{sgn}\Big(-\mathcal{D}(\bar{x}^k, d_{SD}(\bar{x}^k)) + \mathcal{D}(x^k, d_{SD}(\bar{x}^k))\Big)$. It follows from the definitions of $\bar{x}^k$ and $d(\bar{x}^k)$, and Lemma 2 that

$$\begin{aligned}
\mathcal{D}(\bar{x}^k, d(\bar{x}^k)) &\leq \mathcal{D}(\bar{x}^k, d_{SD}(\bar{x}^k)) + \frac{|-\mathcal{D}(\bar{x}^k, d_{SD}(\bar{x}^k)) + \mathcal{D}(x^k, d_{SD}(\bar{x}^k))|}{-\mathcal{D}(x^k, d^k)} \mathcal{D}(\bar{x}^k, \chi d^k) \\
&\leq \mathcal{D}(\bar{x}^k, d_{SD}(\bar{x}^k)) + \frac{L\|d_{SD}(\bar{x}^k)\|\|\bar{x}^k - x^k\|}{-\mathcal{D}(x^k, d^k)} |\mathcal{D}(\bar{x}^k, \chi d^k)| \\
&\leq \mathcal{D}(\bar{x}^k, d_{SD}(\bar{x}^k)) + \frac{L\bar{\alpha}_k\|d_{SD}(\bar{x}^k)\|\|d^k\|}{-\mathcal{D}(x^k, d^k)} |\mathcal{D}(\bar{x}^k, \chi d^k)|.
\end{aligned} \tag{35}$$

Therefore, combining (30) and (35), we obtain

$$-(1-c)\mathcal{D}(\bar{x}^k, d_{SD}(\bar{x}^k)) < \frac{L\bar{\alpha}_k\|d_{SD}(\bar{x}^k)\|\|d_k\|}{-\mathcal{D}(x^k, d^k)} |\mathcal{D}(\bar{x}^k, \chi d^k)|, \quad \forall k \in \mathbb{K}_1. \tag{36}$$

By the definition of $\mathcal{D}(\cdot,\cdot)$, there exists $\bar{w}^k \in C$ such that

$$|\mathcal{D}(\bar{x}^k, \chi d^k)| = |\langle JF(\bar{x}^k)(\chi d^k), \bar{w}^k \rangle| \leq \|JF(\bar{x}^k)\|\|d^k\| \leq \bar{c}\|d^k\|, \tag{37}$$

where the inequalities is due to $\|\bar{w}\| = 1$ and (34). By (36), (37) and Lemma 3(b), we find

$$(1-c)\frac{\|d_{SD}(\bar{x}^k)\|^2}{2} < \frac{\bar{c}L\bar{\alpha}_k\|d_{SD}(\bar{x}^k)\|\|d^k\|^2}{-\mathcal{D}(x^k, d^k)}, \quad \forall k \in \mathbb{K}_1,$$

or, equivalently,

$$\bar{\alpha}_k > -\frac{(1-c)}{2\bar{c}L}\frac{\mathcal{D}(x^k, d^k)}{\|d^k\|^2}\|d_{SD}(\bar{x}^k)\|, \quad \forall k \in \mathbb{K}_1.$$

Hence, from the definition of $\bar{\alpha}_k$ and (16), we obtain

$$\alpha_k > -\frac{\mu(1-c)}{2\bar{c}L}\frac{c\mathcal{D}(x^k, d_{SD}(x^k))}{\|d^k\|^2}\|d_{SD}(\bar{x}^k)\|, \quad \forall k \in \mathbb{K}_1,$$

which, combined with Lemmas 3(b) and 8, yields

$$\alpha_k > \frac{\mu c(1-c)}{4\bar{c}L}\frac{\|d_{SD}(x^k)\|^2}{\|d^k\|^2}\|d_{SD}(\bar{x}^k)\| \geq \frac{\mu c(1-c)}{4\bar{c}L}\left(1+\frac{L(1-c)}{L_0}\right)^{-2}\|d_{SD}(\bar{x}^k)\|, \quad \forall k \in \mathbb{K}_1. \tag{38}$$

From Lemma 8 and (18), it turns out that $\{d^k\}$ is bounded. So, since $\lim_{k\in\mathbb{K}_1}\bar{\alpha}_k = 0$, it follows from the definition of $\bar{x}^k$ that $\lim_{k\in\mathbb{K}_1}\bar{x}^k = x^*$. Therefore, taking limits in (38) for $k \in \mathbb{K}_1$, we conclude that $d_{SD}(x^*) = 0$. Thus, x^* is a K-critical Pareto point of F and the proof is complete. □