An Explicit Spectral Fletcher-Reeves Conjugate Gradient Method for Bi-criteria Optimization

Y. Elboulqe*[1] and M. El Maghri[1]

[1] *Department of Mathematics and Computer Science, Faculty of Sciences Aïn Chock, Hassan II University Casablanca, BP. 5366, Morocco.*

October 12, 2022

Abstract. In this paper we propose a spectral Fletcher-Reeves conjugate gradient-like method (SFRCG) for solving unconstrained bi-criteria minimisation problems without using any technique of scalarization. We suggest an explicit formulae for computing a descent direction common to both criteria. This latter verifies furthermore a sufficient descent property which does not depend on the line search nor on any convexity hypothesis. After proving the existence of a bi-criteria Armijo-type stepsize, global convergence of the proposed algorithm is established. Finally, some numerical results and comparisons with other methods are reported.

Keywords. Multicriteria optimization, Pareto optimality, spectral conjugate gradient method, descent directions, line search.

1 Introduction

Minimization problems with multiple conflicting goals are nowadays often encountered in many scientific areas, where designing optimal strategies is a challenge for decision-making. In general, no single best solution will exist for this type of problems, but a set of efficient solutions, called Pareto equilibria.

Several approaches have been proposed for finding Pareto optima. Most employed strategy is the scalarization (see, e.g., [3, 7, 11]). Here, a family of scalar programs formed by weighting objectives, are solved. This approach, however, presents some inconveniences which can be summarized as follows: the choice of the vector of weights is not known in advance, the order of importance of the objective values, if any, must be predefined, and finally, the inability of the method to explore the landscape of Pareto front.

Recently, multicriteria optimization methods directly derived from nonlinear programming were investigated. This suggests not using any scalarization technique and has the advantage of benefiting from all scalar optimization tools. The idea was first initiated in [24] and next developed in [8, 9, 10, 12, 13, 14, 26].

In this work we focus on developing a new descent method to solve the bi-criteria optimization problem:

$$(P) \quad \underset{x\in\mathbb{R}^n}{\text{Minimize}}\ F(x) := \Big(F_1(x), F_2(x)\Big),$$

where $F_1,\ F_2 : x \in \mathbb{R}^n \longmapsto \mathbb{R}$ are assumed to be of $\mathcal{C}^1$ class on $\mathbb{R}^n$. More precisely, we propose a direct extension of a conjugate gradient (CG) type method to find efficient

*Corresponding author. Email: elboulqeyoussef@gmail.com

solutions of (P) without using any scalarization technique. Our motivation is that CG methods are known by their efficiency, low memory requirements and strong local and global convergence properties.

Technically speaking, CG methods constitute a class of iterative first order algorithms that generate a recurrent sequence $\left(x^k\right)_k$, starting from an initial guess $x^0 \in \mathbb{R}^n$, according to the following relation:

$$x^{k+1} = x^k + t_k d^k, \quad k = 0, 1, 2, \ldots, \tag{1}$$

where t_k is a steplength determined by some line search method and d^k is a descent direction given at each iteration. Let us recall the original CG direction scheme:

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

where the symbol ∇ stands for the vector gradient, here, β_k is an algorithmic parameter particularly chosen to have some good properties. Indeed, when the objective function is quadratic strongly convex, β_k is chosen such that consecutive directions d^k and d^{k-1} are conjugate with respect to the Hessian matrix of F. Hence, in this case, using the exact steplength,

$$t_k = \operatorname*{argmin}_{t>0} \, F(x^k + t d^k),$$

the resulting algorithm finds the minimizer of F in exactly n iterations. In such case, the method is unique and called *linear conjugate gradient method* (LCG) (see, e.g., [25]). The advantages of this method have prompted many specialists to propose extensions to the non-quadratic case, called *nonlinear conjugate gradients methods* (NCG) (see, e.g., [19]). This notably concerns the expression of β_k which differs from one method to another. The well-known expressions of β_k are, for example, Fletcher-Reeves (FR), Ploak-Ribiére-Polyak (PRP), Hestenes-Stiefel (HS), and Dai-Yuan (DY), Conjugate-Descent (CD):

$$\beta_k^{\text{FR}} = \frac{\left\|\nabla F\left(x^k\right)\right\|^2}{\left\|\nabla F\left(x^{k-1}\right)\right\|^2},$$

$$\beta_k^{\text{PRP}} = \frac{\left\langle \nabla F\left(x^k\right), y^k \right\rangle}{\left\|\nabla F\left(x^{k-1}\right)\right\|^2},$$

$$\beta_k^{\text{HS}} = \frac{\left\langle \nabla F\left(x^k\right), y^k \right\rangle}{\left\langle y^k, d^{k-1} \right\rangle},$$

$$\beta_k^{\text{DY}} = \frac{\left\|\nabla F\left(x^k\right)\right\|^2}{\left\langle y^k, d^{k-1} \right\rangle},$$

$$\beta_k^{\text{CD}} = -\frac{\left\|\nabla F\left(x^k\right)\right\|^2}{\left\langle \nabla F\left(x^{k-1}\right), d^{k-1} \right\rangle},$$

where $\langle \cdot, \cdot \rangle$ stands for the Euclidean inner product, $\|\cdot\|$ is the associated norm and $y^k = \nabla F\left(x^k\right) - \nabla F\left(x^{k-1}\right)$.

In terms of line search, it is obvious that an exact stepsize can present computational difficulties, in particular for NCG methods, in the sense that for certain choices of β_k, these methods may not converge (see, e.g., [19]). However, inexact line search techniques have been introduced to avoid cycling. Most employed is the Wolfe technique [29, 30] which has the advantage to ensure that the vector d^k given by (2) is a descent direction at the current iterate x^k. Moreover, with Wolfe's conditions, it is still possible to derive the following Zoutendjik condition being crucial for convergence (see e.g., [19, 25]):

$$\sum_{k\geq 0} \frac{\left\langle \nabla F\left(x^k\right), d^k\right\rangle^2}{\left\|d^k\right\|^2} < \infty. \tag{3}$$

There are also some studies showing that convergence of CG methods is guaranteed under other inexact line searches, for example, the Armijo-type stepsizes introduced in [4] for PRP, FR and CD methods. In [17], another example of Armijo-type line search procedure is suggested for the Polak-Ribière CG method. Anyway, generally, the descent property in CG methods still depend on the used line search.

It is only recently in the last two decades that appeared a new interesting subclass of CG methods, called *spectral conjugate gradient methods* (SCG) [2, 6, 31]. Their main idea is to combine the spectral gradient method [27] with some basic conjugate gradient methods leading to algorithms with better numerical performances. The combination of the two descent schemes consists of taking

$$d^k = -\theta_k \nabla F\left(x^k\right) + \beta_k d^{k-1}, \qquad k \geq 1, \tag{4}$$

where θ_k is an additional parameter that allows to get the sufficient descent property, i.e.,

$$\left\langle \nabla F\left(x^k\right), d^k\right\rangle \leq -c \left\|\nabla F\left(x^k\right)\right\|^2, \tag{5}$$

for some $c > 0$. So, with this modified scheme, it is obvious that the descent property is guaranteed independently of any line search, however, the convergence still needs an appropriate stepsize but only of the Armijo-type.

In recent, some attempts have been made to extend CG methods to the multicriteria framework. Indeed, the authors in [26] propose a generalization of the NCG scheme with several expressions of β_k. Their study focused on the parameters FR, PRP, HS, DY and CD, whereas in the scheme (2), the authors took as the steepest descent direction the multiobjective one introduced in [24] (see also [12]). The convergence of the resulting algorithms was obtained under the Wolfe conditions extended to vector functions. In the same line, other recent works are [16] taking up the method proposed in [26] but with the Hager-Zhang type parameter β_k, and [15] focusing on the particular quadratic multiobjective case.

To our knowledge, no attempt extending SCG methods to the multiobjective case has been made so far. In fact, in this paper we propose SFRCG method to solve (P) which can be considered as a direct generalization of the method proposed in [31] for scalar optimisation. First, we describe the way to explicitly calculate the SFRCG descent direction by using a special convex combination of the gradients of the two criteria. Note that the latter is different from the one providing the steepest descent direction [12, 24]. Indeed, a numerical counterexample shows that it is not possible to obtain an SFRCG descent direction by means of the combination of gradients used in the steepest descent, because of the sufficient descent property that may fail. Next, we prove that the proposed direction satisfies the sufficient descent property and does not depend on convexity nor on the line search technique. More importantly, we prove that for a general sufficient descent direction, there exists an interval of steplengths satisfying the Armijo condition. Under standard hypothesis, global convergence of the proposed algorithm is proved. Finally, some numerical results and comparisons with other methods are reported.

2 The SFRCG method

2.1 The bi-criteria descent scheme

Throughout the manuscript, for $y = (y_1, \ldots, y_p)$ and $y' = (y'_1, \ldots, y'_p) \in \mathbb{R}^p$, $y < y'$ (resp. $y \leq y'$) means that $y_i < y'_i$ (resp. $y_i \leq y'_i$) for all $i = 1, \ldots, p$, and, $y \lneq y'$ means that $y \leq y'$ but $y \neq y'$. Let us recall the following definitions.

Definition 2.1 *A point $x^* \in \mathbb{R}^n$ is said to be*

- *weak Pareto optimum or weakly efficient for (P), if*

$$\nexists x \in \mathbb{R}^n, \; F(x) < F(x^*);$$

- *locally weak Pareto optimum or locally weakly efficient for (P), if it is weakly efficient on a neighbourhood, that is*

$$\nexists x \in \mathcal{N}(x^*), \; F(x) < F(x^*),$$

  where $\mathcal{N}(x^*)$ is some neighbourhood of x^*.

It is well known that a necessary condition for a point $x^* \in \mathbb{R}^n$ to be locally weakly efficient is that

$$-\operatorname{Int}\left(\mathbb{R}^2\right) \cap \operatorname{Im}\left(JF\left(x^*\right)\right) = \emptyset, \tag{6}$$

where $JF(x)$ is the Jacobian matrix of F at x and $\operatorname{Im}(JF(x^*)) \subseteq \mathbb{R}^2$ is the image set of $JF(x^*)$ (see e.g [12]). Obviously, this condition can be rewritten as

$$\nexists d \in \mathbb{R}^2, \; JF\left(x^*\right) d < 0, \tag{7}$$

which also is equivalent to the optimality condition:

$$\exists \lambda \in [0,1], \; \lambda \left[g_1\left(x^*\right) - g_2\left(x^*\right)\right] + g_2\left(x^*\right) = 0, \tag{8}$$

where $g_i(x) := \nabla F_i(x)$, $i = 1, 2$.

A point $x \in \mathbb{R}^n$ satisfying one of the above conditions is called *Pareto critical* or *stationary* point. Hence, if x is not critical, then there exists $d \in \mathbb{R}^2$ such that

$$JF(x)d < 0, \tag{9}$$

which implies that the vector d is a bi-objective descent direction at x, i.e.,

$$\exists t_d > 0, \; \forall t \in \left]0, t_d\right], \; F(x + td) < F(x). \tag{10}$$

The SFRCG iterative process that we propose to solve (P) follows the scheme below:

$$x^{k+1} = x^k + t_k d^k, \quad k = 0, 1, \ldots, \tag{11}$$

where x^k is the current iterate, t_k is a steplength which will be described in the next section and d^k is the descent direction given explicitly by

$$d^k = \begin{cases} -g^k, & \text{if } k = 0, \\ -\theta_k g^k + \beta_k d^{k-1}, & \text{if } k \geq 1, \end{cases} \tag{12}$$

where

$$g^k = \lambda_k\big(g_1^k - g_2^k\big) + g_2^k, \quad \text{with} \quad g_i^k = g_i\big(x^k\big), \quad i = 1, 2, \tag{13}$$

$$\lambda_k = \begin{cases} 1, & \text{if} \quad \big(0 \le a_k \le -b_k\big) \quad \text{or} \quad \big(a_k < 0 \quad \text{and} \quad a_k \le -2b_k\big), \\ 0, & \text{if} \quad \big(a_k \ge 0 \quad \text{and} \quad b_k > 0\big) \quad \text{or} \quad \big(a_k < 0 \quad \text{and} \quad a_k > -2b_k\big), \\ -\dfrac{b_k}{a_k}, & \text{if} \quad 0 \le -b_k < a_k, \end{cases} \tag{14}$$

$$a_k = \begin{cases} \left\|g_1^0 - g_2^0\right\|^2, & \text{if} \quad k = 0, \\ \dfrac{\left\|g_1^k - g_2^k\right\|^2 \times \left\langle g_2^k - g^{k-1}, d^{k-1}\right\rangle - \left\langle g_1^k - g_2^k, g_2^k\right\rangle \times \left\langle g_1^k - g_2^k, d^{k-1}\right\rangle}{\left\|g^{k-1}\right\|^2}, & \text{if} \quad k \ge 1, \end{cases} \tag{15}$$

$$b_k = \begin{cases} \left\langle g_1^0 - g_2^0, g_2^0\right\rangle, & \text{if} \quad k = 0, \\ \dfrac{\left\langle g_1^k - g_2^k, g_2^k\right\rangle \times \left\langle g_2^k - g^{k-1}, d^{k-1}\right\rangle - \left\|g_2^k\right\|^2 \times \left\langle g_1^k - g_2^k, d^{k-1}\right\rangle}{\left\|g^{k-1}\right\|^2}, & \text{if} \quad k \ge 1, \end{cases} \tag{16}$$

$$\beta_k = \frac{\left\|g^k\right\|^2}{\left\|g^{k-1}\right\|^2}, \tag{17}$$

$$\theta_k = \frac{\left\langle g^k - g^{k-1}, d^{k-1}\right\rangle}{\left\|g^{k-1}\right\|^2}. \tag{18}$$

Let us now prove that the vector d^k defined in (12) satisfies a sufficient descent property. To lighten the notations, we put

$$g^k(\lambda) = \lambda\big(g_1^k - g_2^k\big) + g_2^k, \quad \text{for} \quad \lambda \in \mathbb{R}.$$

Proposition 2.1 *Let x^k be the k^{th} iterate of a sequence generated by the procedure (11)–(18). If $g^k \neq 0$, then*

$$\left\langle g^k(\lambda), d^k\right\rangle \le \left\langle g^k, d^k\right\rangle = -\left\|g^k\right\|^2, \quad \forall \lambda \in [0, 1], \tag{19}$$

or equivalently

$$\left\langle g_i^k, d^k\right\rangle \le \left\langle g^k, d^k\right\rangle = -\left\|g^k\right\|^2, \quad \textit{for} \quad i = 1, 2. \tag{20}$$

Else, i.e., $g^k = 0$, then x^k is a Pareto critical point.

Proof To prove the first part (19), we must first prove that λ_k given by (14) is, for each $k \in \mathbb{N}$, the global minimum of the following scalar optimization problem:

$$(P_k) \quad \min_{\lambda \in [0,1]} f_k(\lambda),$$

where the sequence of functions $(f_k)_{k \in \mathbb{N}}$ is defined by

$$f_k(\lambda) = \frac{1}{2} a_k \lambda^2 + b_k \lambda + c_k, \tag{21}$$

a_k and b_k are given respectively by (15) and (16), and c_k is a real number which can be chosen arbitrarily.

It is clear that the problem (P_k) has at least one optimal solution. Let us prove that $\lambda_k \in \operatorname{argmin}(P_k)$. To this end we perform a case disjunction reasoning. If $a_k > 0$ then the problem is strictly convex and if $a_k < 0$ then it is strictly concave. In the first case, $\lambda_k \in \operatorname{argmin}(P_k)$ iff λ_k satisfies the KKT conditions.

$$\begin{cases} a_k\lambda + b_k - \gamma_1 + \gamma_2 = 0, \\ \gamma_1 \lambda = 0, \\ \gamma_2(\lambda - 1) = 0, \\ \gamma_i \geq 0, \quad i = 1, 2, \\ \lambda \in [0, 1], \end{cases}$$

which obviously are satisfied, at each one of the following three cases, by a unique solution:

$$\begin{cases} \lambda_k = 1, & \gamma_1 = 0, \quad \gamma_2 = -a_k - b_k, \quad \text{if} \quad a_k \leq -b_k, \\ \lambda_k = 0, & \gamma_1 = b_k, \quad \gamma_2 = 0, \quad \text{if} \quad b_k > 0, \\ \lambda_k = -\dfrac{b_k}{a_k}, & \gamma_1 = 0, \quad \gamma_2 = 0, \quad \text{if} \quad 0 \leq -b_k < a_k. \end{cases}$$

While in the second case, the only global minimum of (P_k) is 0 or 1. It is easy to verify that the global minimum is $\lambda_k = 1$, if $a_k \leq -2b_k$; otherwise $\lambda_k = 0$. When $a_k = 0$ and $b_k \neq 0$, the function f_k is affine, so, its minimum solution over $[0, 1]$ is reached in 0 or 1 according to $b_k > 0$ or $b_k < 0$ respectively. Finlay, if $a_k = b_k = 0$, any value in $[0, 1]$ is an minimal solution, in particular $\lambda_k = 1$. Thus, in all cases, we obtain (14).

Let's now prove that, for all $k \in \mathbb{N}$, we have (19). To this end, we perform a reasoning by induction. For $k = 0$, the problem (P_0) is a convex program, then by the minimum principal, $\lambda_0 \in \operatorname{argmin}(P_0)$ iff for all $\lambda \in [0, 1]$,

$$f_0'(\lambda_0)(\lambda - \lambda_0) \geq 0,$$

which is equivalent to

$$(a_0\lambda_0 + b_0)(\lambda - \lambda_0) \geq 0.$$

By (15) and (16) we obtain

$$\left(\left\|g_1^0 - g_2^0\right\|^2 \lambda_0 + \left\langle g_2^0, g_1^0 - g_2^0 \right\rangle\right)(\lambda - \lambda_0) \geq 0,$$

which can be easily written under the following form

$$\left\langle g_1^0 - g_2^0, \lambda_0\left(g_1^0 - g_2^0\right) + g_2^0 \right\rangle (\lambda - \lambda_0) \geq 0.$$

Since, by (13), we have $g^0 = \lambda_0\left(g_1^0 - g_2^0\right) + g_2^0$, then

$$\left\langle \lambda\left(g_1^0 - g_2^0\right), g^0 \right\rangle \geq \left\langle \lambda_0\left(g_1^0 - g_2^0\right), g^0 \right\rangle.$$

Adding the term $\left\langle g_2^0, g^0 \right\rangle$ to the two members of the latter inequality, multiplying them by -1 we obtain and by definition of d^0 we obtain

$$\left\langle \lambda\left(g_1^0 - g_2^0\right) + g_2^0, d^0 \right\rangle \leq \left\langle \lambda_0\left(g_1^0 - g_2^0\right) + g_2^0, d^0 \right\rangle,$$

which, prove the result for $k = 0$.

Suppose now that the result is true up to order $k - 1$ and let us prove it for k. Again by the minimum principal, for all $\lambda \in [0, 1]$, we obtain

$$(a_k\lambda_k + b_k)(\lambda - \lambda_k) \geq 0. \tag{22}$$

In the other hand, we have the following equality:

$$a_k\lambda_k + b_k = -\langle g_1^k - g_2^k, d^k\rangle. \tag{23}$$

Indeed, by the definitions of d^k, θ_k, β_k and g^k, we have

$$\begin{aligned}
-\left\langle g_1^k - g_2^k, d^k\right\rangle &= \left\langle g_1^k - g_2^k, \theta_k g^k - \beta_k d^{k-1}\right\rangle \\
&= \theta_k \left\langle g_1^k - g_2^k, g^k\right\rangle + \beta_k \left\langle g_1^k - g_2^k, d^{k-1}\right\rangle \\
&= \frac{\left\langle g^k - g^{k-1}, d^{k-1}\right\rangle}{\|g^{k-1}\|^2} \times \left\langle g_1^k - g_2^k, g^k\right\rangle - \frac{\left\|g^k\right\|^2}{\|g^{k-1}\|^2} \times \left\langle g_1^k - g_2^k, d^{k-1}\right\rangle \\
&= \frac{\left\langle \lambda_k(g_1^k - g_2^k) + g_2^k - g^{k-1}, d^{k-1}\right\rangle}{\|g^{k-1}\|^2} \times \left\langle g_1^k - g_2^k, \lambda_k(g_1^k - g_2^k) + g_2^k\right\rangle \\
&- \frac{\left\|\lambda_k(g_1^k - g_2^k) + g_2^k\right\|^2}{\|g^{k-1}\|^2} \times \left\langle g_1^k - g_2^k, d^{k-1}\right\rangle \\
&= \left(\lambda_k \frac{\left\langle g_1^k - g_2^k, d^{k-1}\right\rangle}{\|g^{k-1}\|^2} + \frac{\left\langle g_2^k - g^{k-1}, d^{k-1}\right\rangle}{\|g^{k-1}\|^2}\right) \times \left(\lambda_k \left\|g_1^k - g_2^k\right\|^2 + \left\langle g_1^k - g_2^k, g_2^k\right\rangle\right) \\
&- \left(\lambda_k^2 \frac{\left\|g_1^k - g_2^k\right\|^2}{\|g^{k-1}\|^2} + 2\lambda_k \frac{\left\langle g_1^k - g_2^k, g_2^k\right\rangle}{\|g^{k-1}\|^2} + \frac{\left\|g_2^k\right\|^2}{\|g^{k-1}\|^2}\right) \times \left\langle g_1^k - g_2^k, d^{k-1}\right\rangle \\
&= \lambda_k \left(\frac{\left\|g_1^k - g_2^k\right\|^2 \times \left\langle g_2^k - g^{k-1}, d^{k-1}\right\rangle}{\|g^{k-1}\|^2} - \frac{\left\langle g_1^k - g_2^k, g_2^k\right\rangle \times \left\langle g_1^k - g_2^k, d^{k-1}\right\rangle}{\|g^{k-1}\|^2}\right) \\
&+ \frac{\left\langle g_1^k - g_2^k, g_2^k\right\rangle \times \left\langle g_2^k - g^{k-1}, d^{k-1}\right\rangle}{\|g^{k-1}\|^2} - \frac{\left\|g_2^k\right\|^2 \times \left\langle g_1^k - g_2^k, d^{k-1}\right\rangle}{\|g^{k-1}\|^2} \\
&= a_k\lambda_k + b_k.
\end{aligned}$$

So, by replacing (23) in (22), we easily obtain

$$\left\langle \lambda(g_1^k - g_2^k), d^k\right\rangle \leq \left\langle \lambda_k(g_1^k - g_2^k), d^k\right\rangle.$$

Hence, by adding the term $\left\langle g_2^k, d^k\right\rangle$ in both sides of the above inequality, it follows that

$$\left\langle g^k(\lambda), d^k\right\rangle \leq \left\langle g^k, d^k\right\rangle. \tag{24}$$

On the other hand, using the definitions of d^k, θ_k and β_k, and the hypothesis of recurrence, we obtain

$$\begin{aligned}
\left\langle g^k, d^k\right\rangle &= -\theta_k \left\|g^k\right\|^2 + \beta_k \left\langle g^k, d^{k-1}\right\rangle \\
&= -\frac{\left\langle g^k - g^{k-1}, d^{k-1}\right\rangle}{\|g^{k-1}\|^2} \times \left\|g^k\right\|^2 + \frac{\left\|g^k\right\|^2}{\|g^{k-1}\|^2} \times \left\langle g^k, d^{k-1}\right\rangle \\
&= -\frac{\left\langle g^k, d^{k-1}\right\rangle \times \left\|g^k\right\|^2}{\|g^{k-1}\|^2} - \left\|g^k\right\|^2 + \frac{\left\|g^k\right\|^2 \times \left\langle g^k, d^{k-1}\right\rangle}{\|g^{k-1}\|^2} \\
&= -\left\|g^k\right\|^2.
\end{aligned}$$

The last equality together with (24) show that the result holds also for k. This completes the proof of (19). The second part (20) follows straightforwardly.

The last part of the proposition is obvious because when $g^k = 0$, a convex combination of the two vectors $g_1\left(x^k\right)$ and $g_2\left(x^k\right)$ is systematically null, which according to condition (8), means that x^k is well a Pareto critical point. □

2.2 Computing the steplength

To complete the SFRCG scheme described by (12)–(18), we introduce a bi-objective Armijo type condition to compute a steplength t_k along a descent direction d^k. By the following proposition we show that there exists a steplength satisfying this condition as soon as d^k is a descent direction.

Proposition 2.2 *Let x^k and d^k as defined in (11) and (12) such that $\left\|g^k\right\| \neq 0$. If $\rho_1 \in]0,1[$ and $\rho_2 > 0$, then there exist $t_a > 0$, such that for all $t \in]0, t_a]$ we have*

$$F_i(x^k + td^k) < F_i(x^k) + \rho_1 t \left\langle g^k, d^k \right\rangle - \rho_2 t^2 \left\|d^k\right\|^2, \quad i = 1, 2. \tag{25}$$

Proof Since we assume that F_i to be differentiable, for $i = 1, 2$, we have

$$F_i\big(x^k + td^k\big) = F_i\big(x^k\big) + \big\langle g_i^k, td^k \big\rangle + R_i(td^k), \quad \text{with} \quad \lim_{t\to 0} \frac{\left|R_i(td^k)\right|}{t\left\|d^k\right\|} = 0. \tag{26}$$

Observe that if $\left\|g^k\right\| \neq 0$, then by (19), we have $\left\langle g_i^k, d^k \right\rangle < 0$, hence $d^k \neq 0$. So, since $\rho_1 \in]0,1[$ and $\rho_2 > 0$, we have

$$\frac{-(1-\rho_1)\left\langle g_i^k, d^k\right\rangle}{\|d^k\|} > 0 \quad \text{and} \quad \frac{R_i(td^k)}{t\left\|d^k\right\|} + t\rho_2 \left\|d^k\right\| \underset{t\to 0}{\longrightarrow} 0, \quad i = 1, 2.$$

So, there exist $t_a > 0$ such that for all $t \in]0, t_a]$, we have

$$\left|\frac{R_i(td^k)}{t\left\|d^k\right\|} + t\rho_2 \left\|d^k\right\|\right| < \frac{-(1-\rho_1)\left\langle g_i^k, d^k\right\rangle}{\|d^k\|}, \quad i = 1, 2.$$

Hence we get for all $t \in]0, t_a]$,

$$R_i\big(td^k\big) < -t(1-\rho_1)\big\langle g_i^k, d^k\big\rangle - t^2\rho_2 \left\|d^k\right\|^2, \quad i = 1, 2.$$

Therefore, form the above inequality, (26) and (20), for $i = 1, 2$ and for all $t \in]0, t_a]$, we get

$$\begin{aligned} F_i\big(x^k + td^k\big) &< F_i\big(x^k\big) + t\big\langle g_i^k, d^k\big\rangle + t\rho_2 \left\|d^k\right\|^2 - t(1-\rho_1)\big\langle g_i^k, d^k\big\rangle \\ &= F_i\big(x^k\big) + t\rho_1\big\langle g_i^k, d^k\big\rangle - t^2\rho_2 \left\|d^k\right\|^2 \\ &\leq F_i\big(x^k\big) + t\rho_1\big\langle g^k, d^k\big\rangle - t^2\rho_2 \left\|d^k\right\|^2, \end{aligned}$$

which shows the result. □

Remark 2.1 1. The Armijo-type condition (25) is a direct extension to bi-objective optimization of the one used in [31] for the scalar case. The above proposition ensures the existence of an interval over which we have (25) when a bi-criteria optimization problem is considered. In the scalar case, the condition has been used without proof of the existence, but, we can easily see that the result may be obtained in the same way for problems with one criterion or also with several criteria more than two. Recall that the classical Armijo condition has been also extended and used for vector functions in [12, 24].

2. We claim that, in general, it is not possible to obtain an SFRCG descent direction by using the steepest descent in (12)–(18). As shown in the following counterexample, the sufficient descent property may fail over iterations at a non Pareto critical point. Let us consider the problem FF1 [21], for which we take $\lambda_k \in \underset{\lambda\in[0,1]}{\operatorname{argmin}} \frac{1}{2}\left\|g^k(\lambda)\right\|^2$ in the scheme (12)–(18). In Table 1 we presented the first two iterations of the calculation program.

k	$(x^k)^T$	$(d^k)^T$	$(JF(x^k)d^k)^T$	t_k
0	$(0.1472139, 0.3145435)$	$(0.1783424, -0.0347479)$	$-(0.0330134, 0.0330134)$	7.8125×10^{-2}
1	$(0.1611469, 0.3118288)$	$(0.1686837, -0.0332744)$	$-(0.0008105, 0.0398217)$	2.4414×10^{-3}
2	$(0.1615587, 0.3117476)$	$(0.1685092, -0.0332481)$	$(0.0001081, -0.0400238)$	$-$

Table 1: The first two iterations of the calculation program.

Observe that the sufficient descent property (19) is not satisfied at the second iteration, because $JF(x^2)d^2 \not< 0$, and, since $\min_{\lambda\in[0,1]} \frac{1}{2}\left\|g^2(\lambda)\right\|^2 = 0.0147254 \neq 0$, then any convex combination of g_1^2 and g_2^2 is different from zero, which means that x^2 is not Pareto critical.

2.3 Global convergence of SFRCG

Based on the above results, we describe bellow the complete SFRCG algorithm.

SFRCG algorithm pseudocode with Armijo type line search

Step 0: Initialisation.

- Choose $x^0 \in \mathbb{R}^n$ and fix Armijo's constant $\rho_1 \in]0,1[\,,\ \rho_2 > 0$.

Let $k = 0$.

Step 1: Descent direction.

- Compute λ_k, g^k and d^k according to (14), (13) and (12) respectively.

Step 2: Stopping criteria.

- If $\left\|g^k\right\| = 0$, then stop: x^k is Pareto critical point for (P).

Step 3: Armijo type line search.

- Determine a steplength $t_k \in]0,1]$ such that

$$t_k = \max_{p\in\mathbb{N}} \left\{ \frac{1}{2^p} \ :\ F_i\big(x^k + \frac{1}{2^p} d^k\big) < F_i\big(x^k\big) + \frac{\rho_1}{2^p}\langle g^k, d^k\rangle - \frac{\rho_2}{4^p}\left\|d^k\right\|^2,\ \text{for } i = 1,2 \right\}.$$

Step 4: Updated point.

- Set $x^k := x^k + t_k d^k$, $k = k+1$ and go to Step 1.

Now, we shall prove the global convergence of SFRCG algorithm under the following classical assumption.

Assumption A

1. The level set $\mathcal{L} = \left\{x \in \mathbb{R}^n \ : \ F(x) \leq F\left(x^0\right)\right\}$ is bounded.
2. In some neighbourhood $\mathcal{N} \supset \mathcal{L}$, the gradients of the two criteria are L-Lipschitz continuous, namely, there exists a constant $L > 0$ such that for all $i = 1, 2$, we have

$$\|g_i(x) - g_i(y)\| \leq L\,\|x - y\|\,, \quad \forall x, y \in \mathcal{N}. \tag{27}$$

From now on, we always suppose that conditions in assumption A hold, and that $\left(x^k\right)_{k\in\mathbb{N}}$ is a sequence generated by the SFRCG algorithm.

It is clear that the sequence $\left(F\left(x^k\right)\right)_{k\in\mathbb{N}}$ is decreasing in the sense that $F\left(x^{k+1}\right) < F\left(x^k\right)$ for all $k \in \mathbb{N}$, so $\left(x^k\right)_{k\in\mathbb{N}} \subseteq \mathcal{L}$.

Before showing the main convergence result, we prove the next important lemma.

Lemma 2.1 *(i) It holds that*

$$\sum_{k\geq 0} t_k \left\|g^k\right\|^2 < +\infty \qquad \text{and} \qquad \sum_{k\geq 0} t_k^2 \left\|d^k\right\|^2 < +\infty. \tag{28}$$

(ii) There exists a constant $c > 0$ such that for all k sufficiently large, we have

$$t_k \geq c\,\frac{\left\|g^k\right\|^2}{\left\|d^k\right\|^2}, \tag{29}$$

(iii) The following Zoutendijk-type condition holds

$$\sum_{k\geq 0} \frac{\left\|g^k\right\|^4}{\left\|d^k\right\|^2} < +\infty. \tag{30}$$

Proof

(i) From the steplength condition and (20), for all $k \in \mathbb{N}$, we have

$$F_i\big(x^k + t_k d^k\big) < F_i\left(x^k\right) - \rho_1 t_k \left\|g^k\right\|^2 - \rho_2 t_k^2 \left\|d^k\right\|^2 \quad i = 1, 2,$$

hence

$$F_i\big(x^k + t_k d^k\big) < F_i\big(x^0\big) - \sum_{l=0}^{k} \left(\rho_1 t_l \left\|g^l\right\|^2 + \rho_2 t_l^2 \left\|d^l\right\|^2\right) \quad i = 1, 2.$$

So, by combining the two above inequalities for $i = 1, 2$, we obtain

$$\frac{1}{2}\sum_{i=1}^{2} \left[F_i\big(x^0\big) - F_i\big(x^k + t_k d^l\big)\right] > \sum_{l=0}^{k} \left(\rho_1 t_l \left\|g^l\right\|^2 + \rho_2 t_l^2 \left\|d^l\right\|^2\right) > 0.$$

The level set $\mathcal{L}$ is compact, then by continuity, the functions F_i, for $i = 1, 2$, are bounded over $\mathcal{L}$. So, since $\left(x^k\right)_{k\in\mathbb{N}} \subset \mathcal{L}$, then there exists $M > 0$, for all $k \in \mathbb{N}$

$$\frac{1}{2}\sum_{i=1}^{2} \left[F_i\big(x^0\big) - F_i\big(x^k + t_k d^k\big)\right] < M,$$

which implies that

$$\sum_{l\geq 0}\left(\rho_1 t_l \left\|g^l\right\|^2+\rho_2 t_l^2\left\|d^l\right\|^2\right)<+\infty.$$

Since, the two terms of the above sum are positives, then

$$\sum_{l\geq 0} t_l\left\|g^l\right\|^2<+\infty \qquad \text{and} \qquad \sum_{l\geq 0} t_l^2\left\|d^l\right\|^2<+\infty.$$

(ii) Observe that in the line search step, i.e., Step 3 of SFRCG algorithm, $0<t_k\leq 1$. So, if t_k was chosen in the first time in the Armijo-type procedure i.e., $t_k=1$, since $\langle g^k,d^k\rangle=-\left\|g^k\right\|^2$, then, by Cauchy–Schwarz inequality, we will have $\left\|g^k\right\|\leq\left\|d^k\right\|$, the result is then satisfied with $c=1$. Else, by the line search step, $2t_k$ does not satisfies the Armijo-type condition (25), that means that there exists $i_0\in\{1,2\}$ such that

$$F_{i_0}\left(x^k+2t_kd^k\right)-F_{i_0}\left(x^k\right)>2\rho_1 t_k\langle g^k,d^k\rangle-4\rho_2 t_k^2\left\|d^k\right\|^2,$$

which, using (19), is equivalent to

$$F_{i_0}\left(x^k+2t_kd^k\right)-F_{i_0}\left(x^k\right)>-2\rho_1 t_k\left\|g^k\right\|^2-4\rho_2 t_k^2\left\|d^k\right\|^2. \tag{31}$$

In the other hand, for all $\gamma\in[0,1]$, we have

$$\left\|x^k-\left(x^k+2\gamma t_kd^k\right)\right\|=2\gamma t_k\left\|d^k\right\|\leq 2t_k\left\|d^k\right\|,$$

and by (28), we obtain $\lim_{k\to\infty} t_k\left\|d^k\right\|=0$. So, since $x^k\in\mathcal{L}\subset\mathcal{N}$, it follows, for k sufficiently large, that

$$x^k+2\gamma t_kd^k\in\mathcal{N},\ \text{ for all } \gamma\in[0,1].$$

Hence, by the mean value theorem, (27) and (20), there exist $\gamma_k\in\,]0,1[$ such that

$$\begin{aligned}
F_{i_0}\left(x^k+2t_kd^k\right)-F_{i_0}\left(x^k\right)&=2t_k\langle g_{i_0}\left(x^k+2\gamma_kt_kd^k\right),d^k\rangle\\
&=2t_k\langle g_{i_0}^k,d^k\rangle+2t_k\langle g_{i_0}\left(x^k+2\gamma_kt_kd^k\right)-g_{i_0}^k,d^k\rangle\\
&\leq 2t_k\langle g_{i_0}^k,d^k\rangle+2t_k\left\|g_{i_0}\left(x^k+2\gamma_kt_kd^k\right)-g_{i_0}^k\right\|\left\|d^k\right\|\\
&=2t_k\langle g_{i_0}^k,d^k\rangle+4L\gamma_kt_k^2\left\|d^k\right\|^2\\
&<2t_k\langle g_{i_0}^k,d^k\rangle+4Lt_k^2\left\|d^k\right\|^2\\
&\leq 2t_k\langle g^k,d^k\rangle+4Lt_k^2\left\|d^k\right\|^2\\
&=-2t_k\left\|g^k\right\|^2+4Lt_k^2\left\|d^k\right\|^2.
\end{aligned}$$

Now, by substituting the last inequality into (31), we get

$$t_k>\frac{(1-\rho_1)\left\|g^k\right\|^2}{2\left(L+\rho_2\right)\|d^k\|^2}.$$

By letting $c=\min\left\{1,\dfrac{1-\rho_1}{2\left(L+\rho_2\right)}\right\}$, we get the result of the assertion.

(iii) This assertion follows immediately using both (i) and (ii). □

We now establish the global convergence theorem for SFRCG algorithm.

Theorem 2.1 *It holds that*

$$\liminf_{k\to\infty} \left\|g^k\right\| = 0.$$

Proof For the sake of contradiction, we suppose that the conclusion is not true. Then there exists a constant $\varepsilon > 0$ such that

$$\left\|g^k\right\| \geq \varepsilon, \quad \forall k \in \mathbb{N}. \tag{32}$$

From (12) we get that

$$\begin{aligned}
\left\|d^k\right\|^2 &= \theta_k^2 \left\|g^k\right\|^2 - 2\theta_k \langle g^k, \beta_k d^{k-1}\rangle + \beta_k^2 \left\|d^{k-1}\right\|^2 \\
&= \theta_k^2 \left\|g^k\right\|^2 - 2\theta_k \langle g^k, d^k + \theta_k g^k\rangle + \beta_k^2 \left\|d^{k-1}\right\|^2 \\
&= \theta_k^2 \left\|g^k\right\|^2 - 2\theta_k \langle g^k, d^k\rangle - 2\theta_k^2 \left\|g^k\right\|^2 + \beta_k^2 \left\|d^{k-1}\right\|^2 \\
&= \beta_k^2 \left\|d^{k-1}\right\|^2 - 2\theta_k \langle g^k, d^k\rangle - \theta_k^2 \left\|g^k\right\|^2.
\end{aligned}$$

Dividing both sides of this equality by $\left\|g^k\right\|^4 = \langle g^k, d^k\rangle^2$, we get from (19), (32) and (17) that

$$\begin{aligned}
\frac{\left\|d^k\right\|^2}{\|g^k\|^4} = \frac{\left\|d^k\right\|^2}{\langle g^k, d^k\rangle^2} &= \beta_k^2 \frac{\left\|d^{k-1}\right\|^2}{\langle g^k, d^k\rangle^2} - \frac{2\theta_k}{\langle g^k, d^k\rangle} - \theta_k^2 \frac{\left\|g^k\right\|^2}{\langle g^k, d^k\rangle^2} \\
&= \left(\frac{\left\|g^k\right\|^2}{\|g^{k-1}\|^2}\right)^2 \times \frac{\left\|d^{k-1}\right\|^2}{\|g^k\|^4} + \frac{2\theta_k}{\|g^k\|^2} - \frac{\theta_k^2}{\|g^k\|^2} \\
&= \frac{\left\|d^{k-1}\right\|^2}{\|g^{k-1}\|^4} - \frac{1}{\|g^k\|^2} \times \left(\theta_k^2 - 2\theta_k + 1 - 1\right) \\
&= \frac{\left\|d^{k-1}\right\|^2}{\|g^{k-1}\|^4} - \frac{(\theta_k - 1)^2}{\|g^k\|^2} + \frac{1}{\|g^k\|^2} \\
&\leq \frac{\left\|d^{k-1}\right\|^2}{\|g^{k-1}\|^4} + \frac{1}{\|g^k\|^2} \\
&\leq \sum_{l=1}^{k} \frac{1}{\|g^l\|^2} \\
&\leq \frac{k}{\varepsilon^2}.
\end{aligned}$$

The last inequalities implies

$$\sum_{k\geq 1} \frac{\left\|g^k\right\|^4}{\|d^k\|^2} \geq \varepsilon^2 \sum_{k\geq 1} \frac{1}{k} = \infty,$$

which contradicts Zoutendijk type condition (30). The proof is then complete. ☐

3 Numerical experiments

This section reports some numerical experiments on the proposed method. Comparisons was made between SFRCG method, multiobjective steepest descent method (SD) in [12, 24] and FR multiobjective conjugate gradient method proposed in [26] that we designed simply by FR. For SFRCG and SD methods, we used our own codes implemented in SCILAB 6.1.0, as for the FR method, we used the code written in Fortran 90 language by its authors and freely available on their website: https://lfprudente.ime.ufg.br/. All codes are executed on the same machine equipped with 1.90 GHz Intel(R) Core(TM) i5 CPU and 16 Go memory.

Eighteen well-known unconstrained bi-criteria test problems that we think are sufficient to reflect essential aspects of the compared methods are experimented (see Tables 2–3). To make a correct comparison between methods, all these problems were solved with the same selected initial population of 200 individuals (starting points), and, for the three algorithms, we considered the same stopping criterion based on the gradient parts of the CG directions. More precisely, for SFRCG method, the stopping condition is that $\left\|g^k\right\|^2 < 10^{-6}$, where the vector g^k is the special convex combination of the gradients of the two criteria defined by (13). As for SD and FR methods, the stopping criteria is $\left\|v(x^k)\right\|^2 < 10^{-6}$, where $v(x^k)$ is the multiobjective steepest descent direction which is also a special convex combination of the two gradients (see [12, 24, 26] for more details). During the numerical experiments, for SFRCG and SD methods we considered respectively $\rho_1 = 10^{-3}$, $\rho_2 = 10^{-8}$ and $\beta = 0.25$ as Armijo's constants, and we started the Armijo procedures with the initial guess $t_0 = 1$.

In order to analyse the obtained graphical representations and to have convenient comparisons between the methods, two essential factors are considered: the convergence of approximate sets towards the exact Pareto front, and the diversity of solutions. These two tasks can be measured with many performance metrics suggested in the literature. Here, we have opted for the three performance measures: Purity metric (P), Hypervolume metric (HV), and the Generational distance (GD). The performances of the three algorithms, relative to Iter, Feval, CPU time, P, HV and GD was evaluated using the profiles of Dolan and Moré [5] (see also [10] for more details about the metric performances and there evaluation).

Tables 2–3 list the numerical results for the three compared methods, where in the first columns we find the names of the considered problems, their references, the number of variable in each problem designed by n as well as the interval where the initial points x^0 was chosen; in the second ones we find, for each test problem, the average number of iterations "iter", the average number of evaluations of vector functions "Feval", the average CPU time in seconds and the obtained performance measurements (see the above abbreviations).

In figure 1 we represented the performance profiles associated to the results of the numerical experiments. Recall that a performance profile is the graphical representation of the (cumulative) distribution function of measures ρ, obtained by a solver on a set of problems with respect to a performance metric (here Iter, Feval, CPU, HV, P or GD).

Note that $\rho(1)$ gives us the largest number of problems among the best solved by a solver according to the analysed performance. However, a value $\rho(\alpha)$ attaining 1 means that all the problems have been solved by the solver at the threshold α. Thus, the best overall performance of a solver is the one that reaches values close to 1 quickly, i.e., for small values of α.

Observe from Fig.1 that our method gives good results with respect to all performance metrics. We can see that the best values $\rho(\alpha)$ could be reached at a fairly small α. By comparing the three methods, it can be seen from the performance profiles of Iter, Eval and CPU that SFRCG and SD methods are much better than FR in term of speed (see also Tables 2–3). In term of convergence, one can see that all the three methods have very good P and GD scores with a slight superiority of SFRCG and SD on FR (see also SLC1 problem in table 3). As for the dispersion of the solutions, the obtained values of HV are very close, which means that the three methods perform well. Recall that the advantage of the proposed method is that it does not require solving any auxiliary optimization problem, the descent is sufficient regardless of the line search, and therefore the possibility of using any type of step. In particular, we used the Armijo step length because of its simplicity.

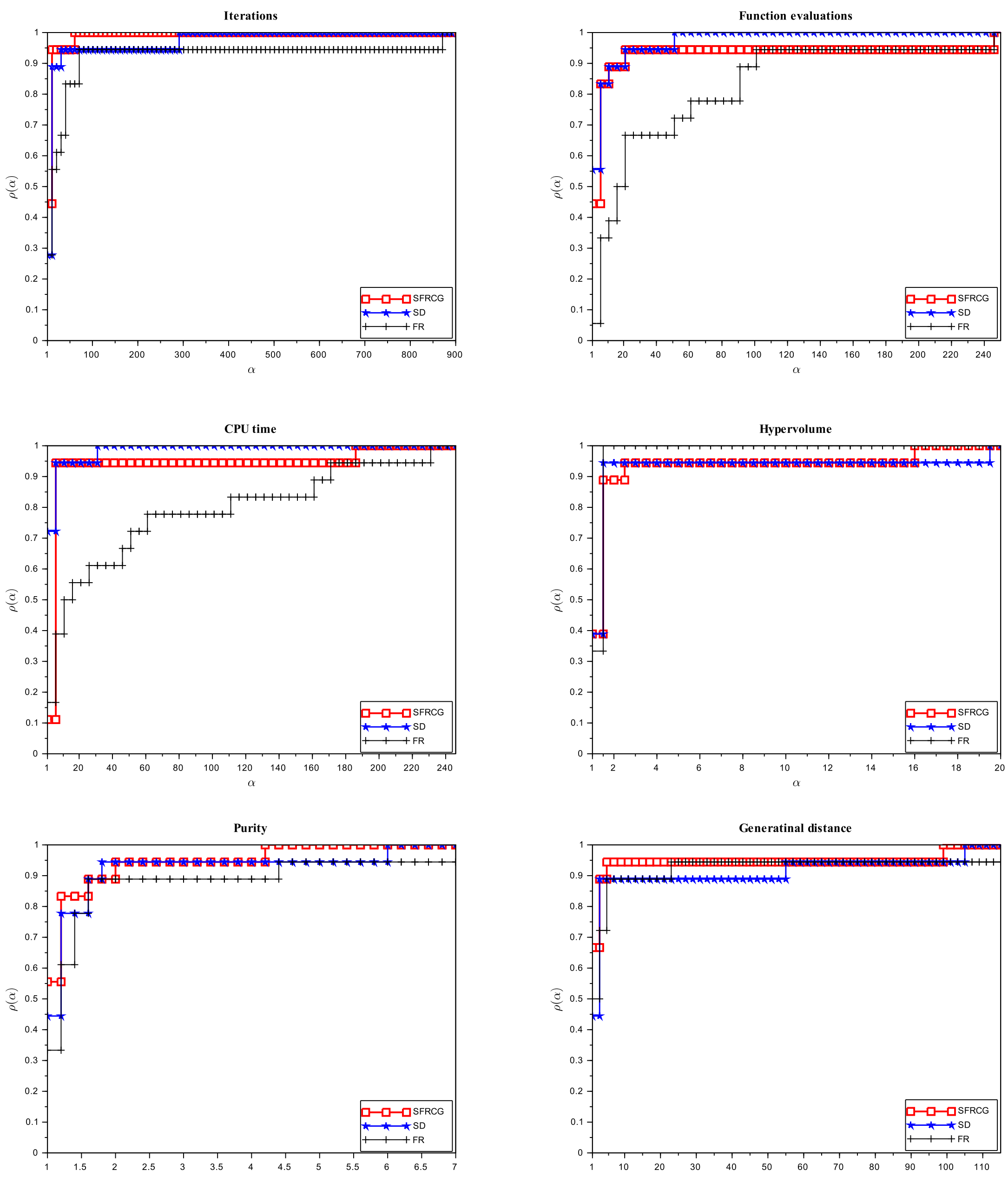

Figure 1: Performance profiles.

Problem		SFRCG	SD	FR
JOS1, [21], $n = 100$, $x^0 \in [-10^4, 10^4]^n$	Iter	7.775	7.77	**1**
	Feval	358.65	358.42	**20**
	CPU	0.0452668	0.0351655	**0.0107031**
	HV	13.157193	**13.177608**	**13.177608**
	P	0.995	**1**	**1**
	GD	0.0000005	**0**	**0**
Lov1, [22], $n = 2$, $x^0 \in [-100, 100]^n$	Iter	3.54	3.24	**2.755**
	Feval	11.24	**10.72**	22.21
	CPU	0.002472	0.0017846	**0.011094**
	HV	**198.34242**	198.0624	197.75275
	P	**0.995**	0.935	0.65
	GD	**0.0000054**	0.0000097	0.0000175
SLC2, [28], $n = 100$, $x^0 \in [-100, 100]^n$	Iter	20.03	**8.575**	26
	Feval	91.585	**39.575**	219
	CPU	0.0379995	**0.0164808**	0.0603125
	HV	**3742.6937**	3679.5183	3712.8274
	P	**1**	**1**	**1**
	GD	**0**	**0**	**0**
SP1, [21], $n = 2$, $x^0 \in [-100, 100]^n$	Iter	8.98	14.01	**6.92**
	Feval	**32.62**	50.085	58.935
	CPU	**0.0087742**	0.0099115	0.0146875
	HV	**14.099308**	14.070517	14.037777
	P	**1**	0.6	0.63
	GD	**0**	0.0000542	0.0000221
AP3, [1], $n = 2$, $x^0 \in [-100, 100]^n$	Iter	**36.83**	1133.29	401
	Feval	**198.815**	10062.245	3249
	CPU	**0.0423827**	1.2859552	0.9260156
	HV	**0.7788456**	0.7781101	0.7787146
	P	0.785	0.505	**0.8**
	GD	0.0048929	0.0037177	**0.0013139**
Far1, [21], $n = 2$, $x^0 \in [-1, 1]^n$	Iter	**18.46**	21.13	1150
	Feval	175.85	**98.44**	8717
	CPU	12.646504	5.6076496	**2.6012499**
	HV	3.5538702	3.5412365	**3.6975253**
	P	**0.27**	0.25	0.2
	GD	0.0103845	0.0107923	**0.0068857**
FF1, [21], $n = 2$, $x^0 \in [-1, 1]^n$	Iter	**24.08**	30.34	845
	Feval	**63.08**	69.22	6365
	CPU	0.0196799	**0.0158182**	2.6318750
	HV	0.0613435	**0.061456**	0.0613647
	P	**0.99**	0.995	0.97
	GD	3.557D-09	**2.734D-09**	6.756D-08
Hil1, [20], $n = 2$, $x^0 \in [0, 1]^n$	Iter	**12.55**	15.575	470
	Feval	**76.625**	81.03	3558
	CPU	0.021689	**0.0187574**	0.88031250
	HV	**1.7247925**	1.4419524	1.6256685
	P	**0.97**	0.92	0.955
	GD	**0.0036514**	0.0063642	0.0047187
Lov3, [22], $n = 2$, $x^0 \in [-100, 100]^n$	Iter	2.745	**2.615**	6
	Feval	9.325	**8.845**	48
	CPU	0.0028814	**0.0019921**	0.0107031
	HV	51051472	**51053274**	51053253
	P	**0.765**	0.76	0.755
	GD	**0.8356897**	0.8378562	0.8381989

Table 2: Numerical results obtained for the considered problems.

		SFRCG	SD	FR
Lov4, [22], $n = 2$, $x^0 \in [-100, 100]^n$	Iter	2.885	2.575	**2**
	Feval	12.62	**11.36**	21
	CPU	0.0037755	0.0028973	**0.0018272**
	HV	**979.66082**	979.6242	979.5777
	P	**0.99**	**0.99**	0.94
	GD	**5.048D-08**	0.0000002	0.0000034
MLF2, [21], $n = 2$, $x^0 \in [-100, 100]^n$	Iter	35.5	**22.15**	1558
	Feval	381.725	**51.405**	12483
	CPU	0.0633905	**0.0142849**	3.2853906
	HV	0.5287019	0.4374676	**8.4546018**
	P	**0.4**	0.37	0.33
	GD	0.0064874	0.0069386	**0.0281064**
MMR1, [23], $n = 2$, $x^0 \in [0, 1]^n$	Iter	**4.46**	7.975	42
	Feval	**29.12**	31.745	347
	CPU	0.0078291	**0.0071319**	0.0921875
	HV	1.112632	1.173147	**1.2767006**
	P	0.205	0.355	**0.375**
	GD	0.0184976	0.0175282	**0.0133341**
MMR5, [23], $n = 100$, $x^0 \in [-5, 5]^n$	Iter	**156.18**	160.985	1730
	Feval	1701.82	**848.835**	12222
	CPU	0.1065144	**0.0458744**	7.3207030
	HV	0.333008	0.3267731	**0.3788657**
	P	0.205	0.145	**0.845**
	GD	0.0029198	0.0031034	**0.00003**
MOP2, [21], $n = 2$, $x^0 \in [-1, 1]^n$	Iter	**7.035**	9.61	160
	Feval	21.925	25.03	1233
	CPU	0.0077907	**0.0057695**	0.3375781
	HV	0.300136	**0.3004738**	0.3001836
	P	**1**	**1**	0.995
	GD	**0**	**0**	4.722D-10
MOP3, [21], $n = 2$, $x^0 \in [-\pi, \pi]^n$	Iter	7.825	**7.465**	26
	Feval	40.805	**31.67**	200
	CPU	0.0147368	**0.0118996**	0.0925781
	HV	**358.1166**	343.87881	347.1766
	P	0.435	**0.495**	0.38
	GD	0.2438789	**0.2335627**	0.2517145
SK2, [21], $n = 4$, $x^0 \in [-10, 10]^n$	Iter	1987.64	**36.575**	1299
	Feval	25399.39	**103.405**	9129
	CPU	4.532554	**0.0245509**	2.7001562
	HV	1193.6865	1260.6049	**1293.3774**
	P	**0.02**	**0.02**	0.03
	GD	168.361	175.00902	**167.50213**
SLC1, [28], $n = 2$, $x^0 \in [-5, 5]^n$	Iter	5.43	5.43	**2**
	Feval	**11.86**	**11.86**	19
	CPU	0.00244	**0.0019722**	0.0023438
	HV	**83.51117**	**83.51117**	77.567641
	P	**1**	**1**	0
	GD	**0**	**0**	0.0127926
VU1, [21], $n = 2$, $x^0 \in [-3, 3]^n$	Iter	932.055	**912.46**	8882
	Feval	8721.685	**3556.01**	71061
	CPU	0.5392446	**0.201866**	9.1066408
	HV	29.248682	**29.256671**	28.748468
	P	0.985	**0.99**	0.23
	GD	**0.0003088**	**0.0003088**	0.0008172

Table 3: Numerical results obtained for the considered problems (continued).

Declaration of competing interest

The authors declare that they have no conflict of interest.

References

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

[2] E. Birgin and J. Martínez, A spectral conjugate gradient method for unconstrained optimization, Appl. Math. Optim. 43 (2001), pp. 117–128.

[3] Y. Collette, P. Siarry, Optimisation Multiobjectif. Eyrolles, Paris (2002).

[4] Y.H. Dai, Conjugate gradient methods with Armijo-type line searches, Acta Mathematicae Applicatae Sinica, English Series, 18 (2002), pp. 123–130.

[5] E. D. Dolan and J. J. Moré, Benchmarking optimization software with performance profiles. Mathematical Programming, 91 (2002), pp. 201–213.

[6] S. Du and Y. Chen, Global convergence of a modified spectral FR conjugate gradient method, Applied Mathematics and Computation, 202 (2008), pp. 766–770.

[7] M. Ehrgott, Multicriteria optimization, 2nd Edition. Springer, Berlin (2005).

[8] M. El Maghri and Y. Elboulqe, Reduced Jacobian method. Journal of Optimization Theory and Applications, 179 (2018), pp. 917–943.

[9] M. El Maghri and Y. Elboulqe, Correction to: reduced Jacobian method. Journal of Optimization Theory and Applications, 187 (2020), pp. 304.

[10] M. El Maghri and Y. Elboulqe, A reduced Jacobian scheme with full convergence for multicriteria optimization. https://optimization-online.org/2022/05/8926/

[11] J. Figueira, S. Greco and M. Ehrgott, Multiple Criteria Decision Analysis, State of The Art Surveys. Springer, New York (2005).

[12] J. Fliege and B.F Svaiter, Steepest descent methods for multicriteria optimization. Mathematical Methods of Operations Research, 51 (2000), pp. 479–494.

[13] J. Fleige, L.M.G Drummond and B.F. Svaiter, Newton's method for multiobjective optimisation. SIAM Journal on Optimization, 20 (2009), 602–626.

[14] E.H. Fukuda and L.M.G. Drummond, A survey on multiobjective descent methods, Pesquisa Operacional 34 (2014), pp. 585–620.

[15] E.H. Fukuda, L.M. Graña Drummond and A.M. Masuda, A conjugate directions-type procedure for quadratic multiobjective optimization, Optimization, https://doi.org/10.1080/02331934.2021.1914034

[16] M.L.N. Gonçalves and L.F. Prudente, On the extension of the Hager–Zhang conjugate gradient method for vector optimization, Computational Optimization and Applications, 76 (2020), pp. 889–916.

[17] L. Grippo and S. Lucidi, A globally convergent version of the Polak-Ribière conjugate gradient method, Mathematical Programming, 78 (1997), pp. 375–391.

[18] L. Guanghui, H. Jiye and Y. Hongxia, Global convergence of the Fletcher-Reeves algorithm with inexact linesearch, Applied Mathematics, 10 (1995), pp. 75–82.

[19] W. W. Hager and H. Zhang, A survey of nonlinear conjugate gradient methods, Pac. J. Optim., 2 (2006), pp. 35–58.

[20] C. Hillermeier, Generalized homotopy approach to multiobjective optimization, J. Optimization Theory and Applications, 110 (2001), pp. 557–583.

[21] S. Huband, P. Hingston, L. Barone and L. While, A review of multiobjective test problems and a scalable test problem toolkit, IEEE Transactions on Evolutionary Computation, 10 (2006), pp. 477–506.

[22] A. Lovison, Singular continuation: Generating piecewise linear approximations to Pareto sets via global analysis, SIAM Journal on Optimization, 21 (2011), pp. 463–490.

[23] E. Miglierina, E. Molho and M. Recchioni, Box-constrained multi-objective optimization: A gradient-like method without "a priori" scalarization, European Journal of Operational Research, 188 (2008), pp. 662–682.

[24] H. Mukai, Algorithms for multicriterion optimization. IEEE Transactions on Automatic Control, 25 (1980), pp. 177–186.

[25] J. Nocedal and S. J. Wright, Numerical Optimization, 2nd Edition, Springer Series in Operations Research and Financial Engineering, Springer, New York (2006).

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

[27] M. Raydan, The Barzilain and Borwein gradient method for the large unconstrained minimization problem, SIAM Journal on Optimization 7 (1997) pp. 26–33.

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

[29] P. Wolfe, Convergence conditions for ascent methods, SIAM Review, 11 (1969), pp. 226–235.

[30] P. Wolfe, Convergence conditions for ascent methods II: Some corrections, SIAM Review, 13 (1971), pp. 185–188.

[31] L. Zhang, W. Zhou and D. Li, Global convergence of a modified Fletcher–Reeves conjugate gradient method with Armijo-type line search, Numer. Math. 104 (2006), pp. 561–572.