Generalized Reduced Jacobian Method

M. El Maghri*[1] and Y. Elboulqe†[2]

[1]*Department of Mathematics and Computer, Faculty of Sciences Aïn Chock, Hassan II University, Casablanca, BP. 5366, Morocco*
[2]*Department of Mathematics, Faculty of Sciences Semlalia, Cadi Ayyad University, Marrakech, BP. 2390, Morocco*

October 17, 2025

Abstract In a recent work, we presented the reduced Jacobian method (RJM) as an extension of Wolfe's reduced gradient method to multicriteria (multiobjective) optimization problems dealing with linear constraints. This approach reveals that using a reduction technique of the Jacobian matrix of the objective avoids scalarization. In the present work, we intend to generalize RJM to handle nonlinear constraints too. In fact, we propose a generalized reduced Jacobian (GRJ) method that extends Abadie-Carpentier's approach for single-objective programs. To this end, we adopt a global reduction strategy based on the fundamental theorem of implicit functions. In this perspective, only a reduced descent direction common to all the criteria is computed by solving a simple convex program. After establishing an Armijo-type line search condition that ensures feasibility, the resulting algorithm is shown to be globally convergent, under mild assumptions, to a Pareto critical (KKT-stationary) point. Finally, experimental results are presented, including comparisons with other deterministic and evolutionary approaches.

Keywords Multicriteria optimization. Pareto optima. Nonlinear programming. Pareto KKT-stationarity. Reduced Jacobian method. Nonlinear constraints.

Mathematics Subject Classification (2020) 90C29 · 90C30 · 90C52

1 Introduction

In many areas, like economics, medicine, design, transportation and so on, we are faced with multicriteria (or multiobjective) optimization problems (MOP), where several functions must be optimized at the same time. Because of the conflicts which may arise between these functions, almost never a single point will minimize or maximize them at once. So the Pareto optimality concept have to be considered. A point is said to be Pareto optimum or efficient solution, if it cannot be improved with respect to a criterion without degrading at least one of the others. Taking account the principle, Pareto solutions are incomparable and no single point can represent all the others. The problem then consists of determining the set of all Pareto solutions or their objective values called the Pareto front or more generally the set of non-dominated solutions.

In recent years, it has been observed that this task has become more tractable with the advent of a new generation of algorithms grounded in the principle of multiobjective search descent directions, which aim to identify a direction that decreases all the criteria simultaneously (e.g. [5, 13, 16, 18, 20, 21, 23, 29, 31, 32, 33, 34]). In most cases, the resulting algorithms are merely direct extensions of well-known nonlinear descent methods, which tackle the given problem without resorting to intermediate transformations or introducing artificial parameters that may be sensitive to the

*Corresponding author. Email: elmaghri@yahoo.com
†Email: elboulqeyoussef@gmail.com

original problem. From a theoretical perspective, convergence toward Pareto-KKT stationarity is well established, and numerically, these approaches have also proven very promising, both in terms of convergence to the true Pareto front and the diversity of non-dominated approximate solutions. For instance, in [16], we proposed a direct extension of the Wolfe reduced gradient method (RGM) [42] (see also [15] for a new RGM variant) to the so-called reduced Jacobian method (RJM) for solving linearly constrained MOPs. We also refer the reader to [18], where we introduced an RJM variant that possesses the full convergence property.

In the present study, we build upon our previous work by focusing on the so-called generalized reduced gradient (GRG) method, originally introduced by Abadie and Carpentier [1], which extends Wolfe's reduction strategy to problems with nonlinear constraints. Accordingly, we propose a novel approach, which we term the generalized reduced Jacobian (GRJ) method, to effectively handle nonlinear constraints in multicriteria optimization. The GRG method is widely recognized as one of the fundamental and practical approaches to nonlinear mathematical programming. Its convergence properties were rigorously established in subsequent works by Smeers [37], Mokhtar-Kharroubi [30], and Schittkowski [36]. As far as we are concerned, our investigations cover both the theoretical and algorithmic aspects of this method. Abadie-Carpentier's reduction strategy, based on the well-known implicit function theorem, is adopted in this work, thereby enabling a complete characterization of Pareto-KKT stationarity through multiobjective reduced descent directions. It is subsequently proven that such directions guarantee simultaneous descent of all criteria and ensure the existence of Armijo-like steplengths, while preserving feasibility throughout the process. From a computational point of view, a simple direction-finding subproblem is introduced to iteratively compute the reduced directions. Under mild assumptions, the resulting GRJ algorithm is proven to be globally convergent to a Pareto KKT-stationary point in the sense of accumulation points. Finally, numerical experiments are reported, illustrating the effectiveness of the GRJ method compared with other methods, including a Zoutendijk-like method [31], an SQP-type approach [22], and the well-known evolutionary method NSGA-II [10].

2 Efficiency and multiobjective descent

We are dealing with the following nonlinearly constrained multicriteria optimization problem:

$$\text{(MOP)} \qquad \text{Min } F(x) \quad \text{subject to} \quad G(x)=0, \;\; a \le x \le b,$$

where $\big[F,G\big] : x \in \mathbb{R}^n \mapsto \big[F(x),G(x)\big] = \big[\big(f_1(x),\ldots,f_r(x)\big),\big(g_1(x),\ldots,g_m(x)\big)\big] \in \mathbb{R}^r \times \mathbb{R}^m$ are two vector-valued functions continuously differentiable on the feasible set denoted by S, and, $a=(a_1,\ldots,a_n)$, $b=(b_1,\ldots,b_n) \in \mathbb{R}^n$. For $y=(y_1,\ldots,y_p) \in \mathbb{R}^p$ and $y'=(y'_1,\ldots,y'_p) \in \mathbb{R}^p$, $y \le y'$ (resp. $y<y'$), iff $y_i \le y'_i$ (resp. $y_i < y'_i$) ($\forall i$), and, $y \lneq y'$ means $y \le y'$ and $y \ne y'$. The usual norm and the usual inner product in $\mathbb{R}^p$ will be respectively denoted by $\|y\| = \sqrt{y \cdot y}$ and $y \cdot y' = y^T y'$, where y^T stands for the transpose of y.

The (MOP) consists in minimizing the vector function F over S in the following Pareto senses:

Definition 2.1 A point $x^* \in S$ is said to be

(a) *weakly efficient* for (MOP), if $\nexists x \in S,\; F(x) < F(x^*)$;

(b) *efficient* (*Pareto optimum*) for (MOP), if $\nexists x \in S,\; F(x) \lneq F(x^*)$;

(c) *properly efficient* (in Henig's sense) for (MOP), if there exists $C \subsetneq \mathbb{R}^r$ a pointed[1] convex cone such that its topological interior $\operatorname{int} C \supset \mathbb{R}^r_+ \setminus \{0\}$, and, $\nexists x \in S,\; F(x) \lneq_C F(x^*)$.[2]

The sets of weakly efficient points, efficient points and properly efficient points will be denoted respectively by E_w, E_e and E_p. To unify the notations, we briefly denote by E_σ the set of all σ-efficient solutions depending on the choice of $\sigma \in \{w,e,p\}$. Recall the following relationships:

$$E_p \subseteq E_e \subseteq E_w. \tag{1}$$

[1] A pointed cone C means that $C \cap -C = \{0\}$.

[2] The relation $y \lneq_C y'$ means that $y' - y \in C \setminus \{0\}$.

The concept of *local σ-efficiency* is defined by replacing S by $S \cap \mathcal{N}(x^*)$ in Definition 2.1, where $\mathcal{N}(x^*)$ is some neighbourhood of x^*. The set of all locally σ-efficient points will be denoted by E_σ^{loc}. The image set $F(E_\sigma)$ (resp. $F(E_\sigma^{loc})$) is called *σ-Pareto front* (resp. *local σ-Pareto front*).

Taking into account the inclusions (1), the determination of the weakly efficient set will include all the other efficient solutions.

Let us denote by $JF(x) = \big(\partial f_j(x)/\partial x_i\big)_{j,i}$ the Jacobian matrix of F at x; the jth line of $JF(x)$ being of course the gradient $\nabla f_j(x)$ of the jth objective f_j at x.

Definition 2.2 Let S be a convex set. The vector mapping $F : S \subset \mathbb{R}^n \to \mathbb{R}^r$ is said to be

(a) *convex* on S, if

$$\forall x, x' \in S,\ \forall \alpha \in [0,1]\,, \quad F(\alpha x + (1-\alpha)x') \leq \alpha F(x) + (1-\alpha)F(x');$$

(b) *strictly convex* on S, if

$$\forall x, x' \in S,\ x \neq x',\ \forall \alpha \in \,]0,1[\,, \quad F(\alpha x + (1-\alpha)x') < \alpha F(x) + (1-\alpha)F(x');$$

(c) *pseudoconvex* on S, if it is pseudoconvex at any $x \in S$, i.e.,

$$\forall x' \in S, \quad F(x') < F(x) \Longrightarrow JF(x)(x'-x) < 0;$$

(d) *strictly pseudoconvex* on S, if it is strictly pseudoconvex at any $x \in S$, i.e.,

$$\forall x' \in S,\ x' \neq x, \quad F(x') \leq F(x) \Longrightarrow JF(x)(x'-x) < 0.$$

Remark 2.1 It is obvious that F is componentwise convex, iff F is convex. But the converse does not hold for pseudoconvexity (see [24, Theorem 9.2.3 and Remark 5.3]). The same should hold true for the strict concepts. As in the scalar case, pseudoconvexity generalizes convexity.

A multiobjective search direction is defined as follows.

Definition 2.3 A vector $d \in \mathbb{R}^n$ is said to be

(a) a *feasible direction* at $x \in S$, if $\exists t_f > 0,\ \forall t \in \,]0, t_f]\,,\ x + td \in S$;

(b) a *tangent direction* to S at $x \in S$, if $\exists (d^k)_k \subset \mathbb{R}^n,\ \exists (t_k)_k \subset \,]0, +\infty[$ such that $\lim\limits_{k \to +\infty} d^k = d$, $\lim\limits_{k \to +\infty} t_k = 0$ and $x + t_k d^k \in S\ \ (\forall k)$;

(c) a *descent direction* of F at $x \in S$, if $\exists t_d > 0,\ \forall t \in \,]0, t_d]\,,\ F(x+td) < F(x)$.

The sets of feasible directions, tangent directions and descent directions are cones which will be denoted respectively by $\mathcal{A}_S(x)$, $\mathcal{T}_S(x)$ and $\mathcal{D}_S(x)$.

A sufficient condition for $d \in \mathcal{D}_S(x)$ is (see [20])

$$JF(x)d < 0. \tag{2}$$

A geometric necessary condition for weak efficiency is obtained using the tangent directions:

Theorem 2.1 *If $x^* \in E_w^{loc}$, then $\nexists d \in \mathcal{T}_S(x^*)$, $JF(x^*)d < 0$.*

Proof Let $d \in \mathcal{T}_S(x^*)$. We can suppose that $d \neq 0$, otherwise the result is trivial. So, by the very definition, $\exists (d^k)_k \longrightarrow d,\ \exists (t_k)_k \searrow 0^+$ such that

$$x^k := x^* + t_k d^k \in S \quad (\forall k). \tag{3}$$

By hypothesis, $\exists \mathcal{N}(x^*)$ a neighbourhood of x^* such that

$$\nexists x \in \mathcal{N}(x^*) \cap S,\ F(x) < F(x^*). \tag{4}$$

From (3), we see that $x^k \in \mathcal{N}(x^*) \cap S$ for k large enough, and from (4), we deduce that

$$\exists K \in \mathbb{N},\ \forall k \geq K,\ \exists j_k \in \{1, \dots, r\},\ f_{j_k}(x^k) \geq f_{j_k}(x^*). \tag{5}$$

Without loss of generality, we can assume that $j_k = j_0$ (a constant index) for k large enough. On the other hand, by differentiability, we have that

$$f_{j_0}\left(x^k\right) = f_{j_0}\left(x^* + t_k d^k\right) = f_{j_0}\left(x^*\right) + t_k \nabla f_{j_0}\left(x^*\right)^T d_k + o\left(t_k d^k\right).$$

It follows, from (5), that for k large enough,

$$t_k \nabla f_{j_0}\left(x^*\right)^T d_k + o\left(t_k d^k\right) \geq 0.$$

Passing to the limit, when $k \nearrow +\infty$, after dividing by $t_k \|d^k\|$, we obtain

$$\nabla f_{j_0}\left(x^*\right)^T d \geq 0,$$

which shows the result of the theorem. □

As is well known, the tangent cone $\mathcal{T}_S(x)$ is introduced to handle nonlinear equality constraints, since the cone $\mathcal{A}_S(x)$ may be empty in such cases (see, e.g., the case where S is a circle). On the other hand, the tangent cone $\mathcal{T}_S(x)$ is always closed and coincides with the closure of $\mathcal{A}_S(x)$ when S is convex. This helps to explain the suitability of using $\mathcal{T}_S(x)$ instead of $\mathcal{A}_S(x)$. Furthermore, although the tangent cone plays a theoretically crucial role for a general feasible set S, its explicit determination often remains challenging in practice. However, when S is explicitly described by equality and/or inequality constraints, the tangent cone can always be expressed in terms of these constraints via the so-called *linearized cone*. For our (MOP), the latter is formulated at a given $x \in S$ as follows:

$$\mathcal{C}_S(x) = \left\{d \in \mathbb{R}^n :\ JG(x)d = 0,\ \forall i \in I_a(x),\ d_i \geq 0,\ \forall i \in I_b(x),\ d_i \leq 0\right\},$$

where $I_a(x) = \{i \in \{1, \dots, n\} : x_i = a_i\}$ and $I_b(x) = \{i \in \{1, \dots, n\} : x_i = b_i\}$ are the index sets of active variables, and, $JG(x)$ is the Jacobian matrix of G at x. In general, one only has that $\mathcal{T}_S(x) \subseteq \mathcal{C}_S(x)$ (see, e.g., [4, Chapter 5]). When equality holds, we say that *Abadie's constraint qualification* ($\mathcal{ACQ}$) is satisfied at x:

$$(\mathcal{ACQ}) \qquad\qquad \mathcal{T}_S(x) = \mathcal{C}_S(x).$$

With the assumption ($\mathcal{ACQ}$), we can state the following explicit conditions of weak efficiency that will be useful in the sequel.

Theorem 2.2 *Assume that* ($\mathcal{ACQ}$) *holds at* $x^* \in S$. *If* $x^* \in E_w^{loc}$, *then the following two equivalent conditions are satisfied:*

(i) There is no tangent descent direction to S at $x^* \in S$ satisfying (2), i.e.,

$$\nexists d \in \mathbb{R}^n : \quad \begin{cases} JF(x^*)d < 0, \\ JG(x^*)d = 0, \\ d_i \geq 0,\ \forall i \in I_a(x^*), \\ d_i \leq 0,\ \forall i \in I_b(x^*). \end{cases}$$

(ii) The point $x^* \in S$ is Pareto KKT-stationary, i.e.,

$$\exists\,(\lambda, u, v, w) \in \mathbb{R}_+^r \setminus \{0\} \times \mathbb{R}^m \times \mathbb{R}_+^n \times \mathbb{R}_+^n : \quad \begin{cases} JF\left(x^*\right)^T \lambda + JG\left(x^*\right)^T u - v + w = 0, \\ v \cdot (x^* - a) = 0, \\ w \cdot (b - x^*) = 0. \end{cases}$$

Conversely, if condition (i) *or* (ii) *is satisfied at $x^* \in S$, F is pseudoconvex (resp. strictly pseudoconvex) at x^* and the kernel $Ker\,JG(x^*)$ contains $\mathcal{N}(0) \cap (S - x^*)$ for some neighbourhood $\mathcal{N}(0)$ of* 0, *then $x^* \in E_w^{loc}$ (resp. $x^* \in E_e^{loc}$).*

Proof *Necessity:* condition (i) is the result of Theorem 2.1 expressed under ($\mathcal{ACQ}$). Equivalence between (i) and (ii) comes directly from Motzkin's alternative theorem (see, e.g., [28]).
Sufficiency: Since it is assumed that $\mathcal{N}(0) \cap (S - x^*) \subseteq \operatorname{Ker} JG(x^*)$, then there exists $\mathcal{N}(x^*)$ a neighbourhood of x^* such that $JG(x^*)(x - x^*) = 0$ for all $x \in \mathcal{N}(x^*) \cap S$. Now, by way of contraposition, suppose that $x^* \notin E_w^{loc}$ (resp. $x^* \notin E_e^{loc}$). Then, there exists $x^0 \in S \cap \mathcal{N}(x^*)$ such that $F(x^0) < F(x^*)$ (resp. $F(x^0) \lneq F(x^*)$). By putting $d = x^0 - x^*$, pseudoconvexity (resp. strict pseudoconvexity) of F at x^* implies that $JF(x^*)d = JF(x^*)\left(x^0 - x^*\right) < 0$. On the other hand, we also have that $JG(x^*)d = JG(x^*)(x^0 - x^*) = 0$ because $x^0 \in \mathcal{N}(x^*) \cap S \subseteq \operatorname{Ker} JG(x^*)$. Moreover,

$$d_i = \begin{cases} x_i^0 - a_i, & \text{if } i \in I_a(x^*), \\ x_i^0 - b_i, & \text{if } i \in I_b(x^*), \end{cases}$$

satisfying $d_i \geq 0$, if $i \in I_a(x^*)$ and $d_i \leq 0$, if $i \in I_b(x^*)$, which under the constraint qualification condition ($\mathcal{ACQ}$), shows the existence of a tangent descent direction of F at x^*. □

Remark 2.2 If, in Theorem 2.2, we replace $\mathcal{N}(0)$ by $\mathbb{R}^n$, then it is easy to see by the same proof that the local efficiency becomes global, in particular, when G is affine.

3 GRJ strategy

Let $A(x) := JG(x) \in \mathbb{R}^{m \times n}$ be the Jacobian matrix of G at $x \in S$ such that $A(x)$ is of full rank $m < n$. Then, there exists a subset $B \subset \{1, \ldots, n\}$, called *basis* of $A(x)$, such that the submatrix $A_B(x)$ is invertible, where (rearranging the columns if necessary) $A(x) = [A_B(x)\ A_N(x)]$ with $A_B(x) = JG_B(x)$, $A_N(x) = JG_N(x)$ and obviously $N = \{1, \ldots, n\} \backslash B$. Rearranging also the components of x we can write $x = (x_B, x_N)$, where x_B (resp. x_N) is the well-known vector of *basic* (resp. *nonbasic*) variables. Since G is assumed to be continuously differentiable and $A_B(x)$ is invertible, then by the implicit function theorem, there exist $V \subset \mathbb{R}^m$ a neighbourhood of x_B, $W \subset \mathbb{R}^{n-m}$ a neighbourhood of x_N and a unique function $\psi :\ W \longrightarrow V$ which is continuously differentiable such that

$$\forall x'_N \in W, \qquad \begin{cases} G\left(\psi(x'_N), x'_N\right) = 0, \\ \dfrac{\partial \psi}{\partial x_N}(x'_N) = -A_B^{-1}(x')A_N(x'). \end{cases} \tag{6}$$

Hence, the feasible set S may be expressed only in terms of nonbasic variables:

$$x \in S \Longleftrightarrow a \leq x = (\psi(x_N), x_N) \leq b.$$

A point $x \in S$ is said to be *nondegenerate*, if there exists B a basis such that $a_B < x_B < b_B$. In this case, B is also said to be a nondegenerate basis for x; otherwise x is *degenerate*. The feasible set S is in turn called nondegenerate, if any $x \in S$ is nondegenerate.

Given a vector $d \in \mathbb{R}^n$ that we partition in the same way to $d = (d_B, d_N)$, then it is clear that

$$JG(x)d = 0 \iff d_B = -A_B^{-1}(x)A_N(x)d_N.$$

So by also partitioning the Jacobian matrix $JF(x) = [JF_B(x)\ JF_N(x)]$, for any d such that $d_B = -A_B^{-1}(x)A_N(x)d_N$, it follows that

$$JF(x)d = U_N(x)d_N,$$

where $U_N(x)$ is an $r \times (n-m)$ matrix, which will be called *generalized reduced Jacobian matrix* of F at x, explicitly given by

$$U_N(x) := JF_N(x) - JF_B(x)A_B^{-1}(x)A_N(x). \tag{7}$$

Observe that the jth row of $U_N(x)$ is nothing more than the well-known reduced gradient of the criterion f_j at x which is given by

$$u_N^j(x) = \frac{\partial f_j}{\partial x_N}(x) - \big(A_B^{-1}(x)A_N(x)\big)^T \frac{\partial f_j}{\partial x_B}(x).$$

Now, we define a *multiobjective reduced descent direction* of F at x as being a nonbasic vector $d_N \in \mathbb{R}^{n-m}$ satisfying:

$$U_N(x)d_N < 0, \quad \forall i \in I_a(x) \cap N,\ d_i \geq 0, \quad \text{and} \quad \forall i \in I_b(x) \cap N,\ d_i \leq 0. \tag{8}$$

The geometric stationarity stated in Theorem 2.2(i) then can be expressed only in terms of nonbasic directions:

Corollary 3.1 *Under ($\mathcal{ACQ}$), a nondegenerate point $x \in S$ is Pareto KKT-stationary for (MOP), iff there is no multiobjective reduced descent direction of F at x.*

The next result shows that given any multiobjective reduced descent direction at a given nondegenerate point, feasibility along this direction with an Armijo-like steplength is guaranteed.

Proposition 3.1 *Let $x \in S$ and ψ be the associated implicit function given by (6), and, let d_N be a multiobjective reduced descent direction of F at x. Let $t \in \mathbb{R}_+$ and define*

$$t_N := \min\left(\left\{\frac{a_i - x_i}{d_i} \ :\ d_i < 0,\ i \in N\right\} \cup \left\{\frac{b_i - x_i}{d_i} \ :\ d_i > 0,\ i \in N\right\}\right). \tag{9}$$

Then,

(i) $t_N > 0$.

(ii) $t \leq t_N \Longleftrightarrow a_N \leq x_N + td_N \leq b_N$.

If furthermore $a_B < x_B < b_B$ (i.e., x is nondegenerate with $B = \{1, \dots, n\} \setminus N$), then

(iii) $\exists t_f \in \,]0, t_N]$, $\forall t \in \,]0, t_f]$,

$$\big(\psi(x_N + td_N), x_N + td_N\big) \in S \quad \textit{and} \quad a_B < \psi(x_N + td_N) < b_B. \tag{10}$$

(iv) $\forall \beta \in \,]0, 1[$ *(fixed)*, $\exists t_a \in \,]0, t_f]$, $\forall t \in \,]0, t_a]$,

$$F\big(\psi(x_N + td_N), x_N + td_N\big) < F(x) + \beta t U_N(x)d_N. \tag{11}$$

Proof (i) It is clear that t_N exists (since $d_N \neq 0$). The positivity of t_N follows by the very definition of d_N. Indeed, if $t_N = (a_i - x_i)/d_i$, then from (8), it follows that $a_i - x_i < 0$, and consequently, $t_N > 0$; otherwise, $t_N = (b_i - x_i)/d_i$, which in accordance with (8), entails that $b_i - x_i > 0$, thereby ensuring $t_N > 0$.

(ii) Let us show the first implication. Let $i \in N$. The assertion is obvious if $d_i = 0$. Now, if $d_i > 0$, then $a_i \leq x_i + td_i$ and $(b_i - x_i)/d_i \geq t_N \geq t$, so that $b_i \geq x_i + td_i$. The case $d_i < 0$ is similar. Conversely, since $d_N \neq 0$, then if $d_i < 0$ ($i \in N$), the hypothesis $a_i \leq x_i + td_i$ implies that $t \leq (a_i - x_i)/d_i$; and if $d_i > 0$, then the hypothesis $x_i + td_i \leq b_i$ implies that $t \leq (b_i - x_i)/d_i$, which shows that $t \leq t_N$.

(iii) By definition of $\psi : W \longrightarrow V$ (see (6)) and the continuity of $t \longmapsto x_N + td_N$ at 0, there exists $t_1 > 0$ such that $x_N + td_N \in W$ for all $t \in [0, t_1]$. It follows by (6) that

$$\forall t \in [0, t_1], \quad G\left(\psi(x_N + td_N), x_N + td_N\right) = 0.$$

On the other hand, by assertion (ii), we have that

$$\forall t \in [0, t_N], \quad a_N \leq x_N + td_N \leq b_N.$$

So, it remains to show that the box-constraint also holds for the basic variables. Indeed, by nondegeneracy assumption and continuity, it is immediate that

$$\exists t_2 > 0, \quad \forall t \in [0, t_2], \quad a_B < \psi(x_N + td_N) < b_B.$$

Thus, by putting $t_f = \min\{t_1, t_2, t_N\}$, we obtain the desired result.

(iv) To prove the Armijo-like inequality (11), let us consider the reduced function $\widetilde{F}(\cdot) := F(\psi(\cdot), \cdot)$ which is continuously differentiable on $W \subset \mathbb{R}^{n-m}$. By using the derivative of ψ given by (6), we immediately deduce the Jacobian of $\widetilde{F}$:

$$J\widetilde{F}(x_N) = JF_N(x) - JF_B(x)A_B(x)^{-1}A_N(x) = U_N(x).$$

By virtue of (iii), $x_N + td_N \in W$ for all $t \in]0, t_f]$. So, by differentiability, it follows that

$$\begin{aligned} \widetilde{F}(x_N + td_N) &= \widetilde{F}(x_N) + tU_N(x)d_N + o(t) \\ &= \widetilde{F}(x_N) + t\beta U_N(x)d_N + t\big[(1-\beta)U_N(x)d_N + \tfrac{1}{t}o(t)\big]. \end{aligned}$$

Since $(1-\beta)U_N(x)d_N < 0$ and $\lim\limits_{t \to 0^+} \dfrac{1}{t}o(t) = 0$, then

$$\exists t_a \in]0, t_f], \ \forall t \in]0, t_a], \ \ (1-\beta)U_N(x)d_N + \frac{1}{t}o(t) < 0,$$

thus implying that

$$\widetilde{F}(x_N + td_N) < \widetilde{F}(x_N) + t\beta U_N(x)d_N.$$

As we have $\widetilde{F}(x_N + td_N) = F\big(\psi(x_N + td_N), x_N + td_N\big)$ and $\psi(x_N) = x_B$, then the result follows straightforwardly. □

After having shown that it is possible to obtain feasible steplengths along reduced directions, it remains for completing our GRJ scheme to show how to determine such reduced descent directions.

4 GRJ search direction

Let us consider first a positive functional $\phi : \mathbb{R} \to \mathbb{R}_+$ satisfying $\phi(t) = 0$ iff $t = 0$; e.g., $\phi_p(t) = \dfrac{|t|^p}{p}$ for some value of $p \in]0, 1]$, or, $\phi_0(t) = \mathbb{1}_{\mathbb{R}^*}$ the characteristic function of $\mathbb{R}^*$ defined by $\mathbb{1}_{\mathbb{R}^*}(t) = 1$ if $t \neq 0$; $\mathbb{1}_{\mathbb{R}^*}(0) = 0$. Let $\lfloor a \rfloor_+ := \max(0, a)$ and $\lfloor a \rfloor_- := \max(0, -a)$ denote, respectively, the positive and negative parts of the scalar a. Then, for $x \in S$, we introduce the following finding-direction subproblem:

$$(\mathcal{P}_x) \qquad \operatorname*{Min}_{\lambda \in \Lambda} \ f(\lambda, x) := \frac{1}{2} \sum_{i \in N} \left(\phi\left(b_i - x_i\right) \Big\lfloor \left(U_N(x)^T \lambda\right)_i \Big\rfloor_-^2 + \phi\left(x_i - a_i\right) \Big\lfloor \left(U_N(x)^T \lambda\right)_i \Big\rfloor_+^2 \right)$$

where $\Lambda = \{(\lambda_1, \ldots, \lambda_r) \in \mathbb{R}_+^r \ : \ \sum_{j=1}^r \lambda_j = 1\}$, x_i (resp. a_i and b_i) is the *ith* component of x_N (resp. a_N and b_N) and $\left(U_N(x)^T \lambda\right)_i$ is the *ith* component of $U_N(x)^T \lambda$.

It is obvious that $(\mathcal{P}_x)$ is a convex continuously differentiable problem and always has an optimal solution in the compact set Λ. Moreover, $(\mathcal{P}_x)$ has simplicial constraints and it is easy to verify that its objective function has a computable gradient explicitly given by

$$\nabla_\lambda f(\lambda, x) = -U_N(x)\delta_N(\lambda, x),$$

where

$$\delta_N(\lambda, x) = \left(\phi\left(b_i - x_i\right) \Big\lfloor \left(U_N(x)^T \lambda\right)_i \Big\rfloor_- - \phi\left(x_i - a_i\right) \Big\lfloor \left(U_N(x)^T \lambda\right)_i \Big\rfloor_+ \right)_{i \in N}. \tag{12}$$

The descent GRJ scheme will consist to take $d_N := \delta_N(\lambda(x), x)$, where $\lambda(x) \in \operatorname{argmin}(\mathcal{P}_x)$. Then, according to (12), $\forall i \in N$,

$$d_i = \begin{cases} -\phi(x_i - a_i)\left(U_N(x)^T\lambda(x)\right)_i, & \text{if} \quad \left(U_N(x)^T\lambda(x)\right)_i > 0, \\ -\phi(b_i - x_i)\left(U_N(x)^T\lambda(x)\right)_i, & \text{else.} \end{cases} \tag{13}$$

The lemma below characterizes the optimal solutions of $(\mathcal{P}_x)$, which will be crucial not only for proving the descent properties of the vector d_N, but also for the convergence analysis of the resulting algorithm. Its proof is omitted, as it is similar to the one of [16, Lemma 3.2].

Lemma 4.1 *$\lambda(x) \in \Lambda$ is an optimal solution of $(\mathcal{P}_x)$, iff*

$$\delta_N(\lambda(x), x) \cdot \left(U_N(x)^T\lambda\right) \leq -2f\left(\lambda(x), x\right) \qquad (\forall \lambda \in \Lambda),$$

or equivalently,

$$\delta_N(\lambda(x), x) \cdot u_N^j(x) \leq -2f\left(\lambda(x), x\right) \qquad (j = 1, \ldots, r).$$

The descent properties are stated as follows:

Proposition 4.1

(i) $d_N = 0 \Longleftrightarrow f(\lambda(x), x) = 0$.

(ii) $d_N \neq 0 \Longrightarrow d_N$ *is a multiobjective reduced descent direction of F at x. If moreover $(\mathcal{ACQ})$ holds and x is nondegenerate, then x is not Pareto KKT-stationary for (MOP).*

(iii) $d_N = 0 \Longrightarrow x$ *is Pareto KKT-stationary for (MOP).*

(iv) The functional $x \mapsto f(\lambda(x), x)$ is continuous on S provided ϕ is continuous.

Proof (i) To show the direct implication, observe that as $\phi \geq 0$, then $d_N = 0$ implies that, $\forall i \in N$,

$$\left\lfloor \phi\left(b_i - x_i\right)\left(U_N(x)^T\lambda(x)\right)_i \right\rfloor_- = \left\lfloor \phi\left(x_i - a_i\right)\left(U_N(x)^T\lambda(x)\right)_i \right\rfloor_+.$$

If we suppose that the last two terms are not equal to zero, we would have

$$\phi\left(b_i - x_i\right)\left(U_N(x)^T\lambda(x)\right)_i < 0 \quad \text{and} \quad \phi\left(x_i - a_i\right)\left(U_N(x)^T\lambda(x)\right)_i > 0,$$

which would mean that $\phi\left(b_i - x_i\right)$ and $\phi\left(x_i - a_i\right)$ have not the same sign contradicting $\phi \geq 0$. Thus, $\forall i \in N$,

$$\left\lfloor \phi\left(b_i - x_i\right)\left(U_N(x)^T\lambda(x)\right)_i \right\rfloor_- = \left\lfloor \phi\left(x_i - a_i\right)\left(U_N(x)^T\lambda(x)\right)_i \right\rfloor_+ = 0,$$

and this shows that $f(\lambda(x), x) = 0$. The reverse implication follows straightforwardly from the positivity of ϕ and (13).

(ii) If $d_N \neq 0$, then by assertion (i), $f(\lambda(x), x) > 0$. It follows, by Lemma 4.1, that

$$d_N \cdot u_N^j(x) \leq -2f(\lambda(x), x) < 0 \qquad (j = 1, \ldots, r).$$

Hence, $U_N(x)d_N < 0$. On the other hand, for $i \in I_a(x) \cap N$, we have that

$$d_i = \phi(b_i - x_i)\left\lfloor \left(U_N(x)^T\lambda(x)\right)_i \right\rfloor_- \geq 0,$$

and, for $i \in I_b(x) \cap N$, we have that

$$d_i = -\phi(a_i - x_i)\left\lfloor \left(U_N(x)^T\lambda(x)\right)_i \right\rfloor_+ \leq 0.$$

This shows, according to the definition (8), that d_N is well a multiobjective reduced descent direction. The second part of this assertion follows straightforwardly from Corollary 3.1.

(iii) Suppose that $d_N = 0$. If we put

$$v_N^* := \left(\left\lfloor \left(U_N(x)^T\lambda(x)\right)_i \right\rfloor_+ \right)_{i\in N} \quad \text{and} \quad w_N^* := \left(\left\lfloor \left(U_N(x)^T\lambda(x)\right)_i \right\rfloor_- \right)_{i\in N},$$

then, by (13) and the property that $\phi(t) = 0$ iff $t = 0$, we obtain

$$(v_N^*, w_N^*) \geq 0, \quad v_N^* \cdot (x_N - a_N) = 0 \quad \text{and} \quad w_N^* \cdot (b_N - x_N) = 0.$$

Clearly, for $\lambda^* := \lambda(x)$, $u^* := -(A_B^{-1}(x))^T JF_B(x)^T\lambda^*$, $v^* := (0, v_N^*)$ and $w^* := (0, w_N^*)$, it holds that $(\lambda^*, u^*, v^*, w^*) \in \mathbb{R}_+^r \setminus \{0\} \times \mathbb{R}^m \times \mathbb{R}_+^n \times \mathbb{R}_+^n$, $JF(x)^T\lambda^* + JG(x)^T u^* - v^* + w^* = 0$ and $v^* \cdot (x - a) = w^* \cdot (b - x) = 0$, which express the Pareto KKT-stationarity for (MOP).

(iv) Since $f(\lambda(x), x)$ is the optimal value of $(\mathcal{P}_x)$, its continuity with respect to x follows from the compactness of the feasible set Λ and the continuity of the function $(\lambda, x) \mapsto f(\lambda, x)$. □

5 GRJ algorithm and its convergence

5.1 The algorithm

The generalized reduced Jacobian algorithm we propose to solve the multicriteria problem (MOP) is described below:

GRJ pseudocode with Armijo line search

Step 0: Initialization.

- Select $x \in S$, fix the Armijo's constant $\beta \in \,]0, 1[$ and choose the functional ϕ.

Step 1: Nondegenerate basis selection.

- Identify a basis B such that $a_B < x_B < b_B$. Set $N = \{1, \ldots, n\} \setminus B$.

Step 2: Generalized reduced Jacobian.

- Compute $U_N(x)$ according to (7).

Step 3: Reduced direction.

- Solve $(\mathcal{P}_x)$ to determine d_N according to (13).

Step 4: Stopping criterion.

- If $\min(\mathcal{P}_x) = 0$, then STOP: x is Pareto KKT-stationary for (MOP).

Step 5: Feasible Armijo line search.

- Compute t_N according to (9).

- Determine, according to (10)-(11), a steplength $t \in \,]0, t_N]$ that satisfies:

$$t = \max_{p\in\mathbb{N}} \left\{ \frac{1}{2^p} \quad : \quad \begin{array}{l} \left(\psi\left(x_N + \frac{1}{2^p}d_N\right), x_N + \frac{1}{2^p}d_N\right) \in S, \text{ and} \\[2ex] F\left(\psi\left(x_N + \frac{1}{2^p}d_N\right), x_N + \frac{1}{2^p}d_N\right) < F(x) + \beta\frac{1}{2^p}U_N(x)d_N \end{array} \right\},$$

where ψ is the implicit function defined in (6).

Step 6: Updated point.

- Set $x := \left(\psi\left(x_N + td_N\right), x_N + td_N\right)$.

Step 7: Degeneracy test.

- If B is a degenerate basis for x, go to Step 1; otherwise go to Step 2.

5.2 Convergence analysis

If the algorithm terminates after a finite number of iterations, then according to Proposition 4.1, it terminates at a Pareto KKT-stationary point. So we will assume that the sequence produced by the algorithm is infinite. We also assume that the bases are chosen in such a way that the following hypothesis is satisfied:

$$(\mathcal{H}) \qquad \left.\begin{cases} x^k \underset{k\in K\subseteq\mathbb{N}}{\longrightarrow} x^* \text{ non Pareto KKT-stationary,} \\ \\ \exists B \text{ basis, } \forall k\in K,\ a_B < x_B^k < b_B. \end{cases}\right\} \Longrightarrow a_B < x_B^* < b_B.$$

Known as the *basis property*, this hypothesis proves to be essential for the global convergence of reduced gradient methods (e.g., [16, 18, 30, 37] [3]). An example provided in [30] shows that, without this hypothesis, the algorithm may fail to converge even if the constraints are linear. However, there are basis selection procedures that ensure the basis property for both linear and nonlinear constraint cases, such as those mentioned in [18].

Theorem 5.1 *Assume that S is nondegenerate, ϕ is chosen to be continuous and derivable at* 0*. Let $(x^k)_{k\in\mathbb{N}}$ be the sequence produced by the GRJ algorithm. Then,*

(i) $x^k \in S$ and $F(x^{k+1}) < F(x^k)$ for all $k\in\mathbb{N}$;

(ii) under the hypothesis ($\mathcal{H}$), any accumulation point x^* of $(x^k)_k$ is a Pareto KKT-stationary point for (MOP), and, $\lim\limits_{k\to+\infty} F(x^k) = F(x^*)$.

Proof (i) This assertion follows directly from Proposition 3.1.
(ii) We need to prove the following lemma.

Lemma 5.1 *If, in addition to the hypotheses of Theorem 5.1, a subsequence $(x^k)_{k\in K}$ produced by the GRJ algorithm converges to x^* with constant bases $B_k = B$ for all $k\in K$, and $\big(\lambda(x^k)\big)_{k\in K}$ converges too, then*

(a) $d_N^k := \delta_N(\lambda(x^k), x^k) \underset{k\in K}{\longrightarrow} d_N^* := \delta_N(\lambda(x^*), x^*)$, where $N = \{1,\dots,n\}\setminus B$;

(b) $\bar t := \inf\limits_{k\in K} t_N^k > 0$, where

$$t_N^k := \min\left(\left\{\frac{a_i - x_i^k}{d_i^k} \ :\ d_i^k < 0,\ i\in N\right\} \cup \left\{\frac{b_i - x_i^k}{d_i^k} \ :\ d_i^k > 0,\ i\in N\right\}\right).$$

Proof (a) Let $\lambda^* = \lim\limits_{\substack{k\in K\\ k\to+\infty}} \lambda(x^k)$, then $\lambda^*\in\Lambda$ since $\big(\lambda(x^k)\big)_{k\in K}\subset\Lambda$ which is closed. According to the hypothesis and Lemma 4.1, for all $k\in K$, we have

$$\delta_N(\lambda(x^k), x^k)\cdot u_N^j(x^k) \le -2f(\lambda(x^k), x^k) \qquad (j = 1,\dots,r),$$

Taking into account that the mappings $(\lambda, x)\longmapsto f(\lambda, x)$ and $(\lambda, x)\longmapsto \delta_N(\lambda, x)$ are continuous, by passing onto the limit, when $k\nearrow +\infty$ in the previous inequality, we obtain that

$$\delta_N(\lambda^*, x^*)\cdot u_N^j(x^*) \le -2f(\lambda^*, x^*) \qquad (j = 1,\dots,r),$$

Since by closeness of S, x^* remains feasible as $(x^k)_k$, this shows applying once again Lemma 4.1 that $\lambda^*\in\operatorname{argmin}(\mathcal{P}_{x^*})$, i.e., $\lambda^* = \lambda(x^*)$, which proves the assertion.

(b) To prove this assertion, we shall proceed by a contradiction way by assuming that $\bar t = 0$ (because $\bar t \ge 0$). Then, it would exist $K_0\subseteq K$ an infinite subset such that $t_N^k \underset{k\in K_0}{\longrightarrow} 0$. Since the

[3] As pointed out in [18], the hypothesis ($\mathcal{H}$) was inadvertently omitted in the proof of [16, Theorem 4.1].

index set N is finite, we can assume (without loss of generality) that one of the two following cases would happen:

$$\exists i_0 \in N, \quad \forall k \in K_0, \quad t_N^k = \frac{a_{i_0} - x_{i_0}^k}{d_{i_0}^k}, \quad d_{i_0}^k < 0,$$

or

$$\exists i_0 \in N, \quad \forall k \in K_0, \quad t_N^k = \frac{b_{i_0} - x_{i_0}^k}{d_{i_0}^k}, \quad d_{i_0}^k > 0.$$

In the first case, following (13), we would have

$$\forall k \in K_0, \quad t_N^k = \frac{1}{(U_N(x^k)^T \lambda(x^k))_{i_0}} \left[\frac{x_{i_0}^k - a_{i_0}}{\phi(x_{i_0}^k - a_{i_0})} \right]. \tag{14}$$

Now, by assertion (a), $d_{i_0}^k := -\phi(x_{i_0}^k - a_{i_0}) \left(U_N(x^k)^T \lambda(x^k)\right)_{i_0} \underset{k \in K_0}{\longrightarrow} d_{i_0}^*$, and, by the contradiction hypothesis, $t_N^k \underset{k \in K_0}{\longrightarrow} 0$. So we would have that the sequence $\left(x_{i_0}^k - a_{i_0}\right)_{k \in K_0}$ also converges to 0; i.e., $x_{i_0}^k \underset{k \in K_0}{\longrightarrow} a_{i_0}$. But this would imply that

$$\phi(x_{i_0}^k - a_{i_0}) \underset{k \in K_0}{\longrightarrow} \phi(x_{i_0}^* - a_{i_0}) = \phi(0) = 0,$$

and, by letting $k \nearrow +\infty$ in (14), we would obtain the contradiction

$$t_N^k \underset{k \in K_0}{\longrightarrow} \frac{1}{(U_N(x^*)^T \lambda^*)_{i_0}} \times \frac{1}{\phi'(0)} \neq 0,$$

The second case is similar. Hence, $\bar{t} > 0$. □

Let us go back to the proof of Theorem 5.1(ii).

By hypothesis, there exists a subsequence $(x^k)_{k \in K} \longrightarrow x^*$ with $K \subseteq \mathbb{N}$ an infinite subset. By closeness of S, x^* remains feasible as $(x^k)_k$. Now, by continuity of F and the assertion (i), it follows that the sequence $\left(F(x^k)\right)_{k \in \mathbb{N}} \longrightarrow F(x^*)$. On the other hand, since the index set B_k is in the finite set $\{1, \dots, n\}$, we can assume (without loss of generality) that $B_k = B$ (hence $N_k = N$) for all $k \in K$. Now, by Armijo's condition: $\forall k \in \mathbb{N}$,

$$F(x^{k+1}) - F(x^k) < \beta t_k U_{N_k}(x^k) d_{N_k}^k < 0.$$

It follows that

$$\lim_{\substack{k \in K \\ k \to +\infty}} t_k U_N(x^k) d_N^k = \lim_{\substack{k \in K \\ k \to +\infty}} t_k U_{N_k}(x^k) d_{N_k}^k = 0. \tag{15}$$

Similarly, as the sequence $\left(\lambda(x^k)\right)_k \subset \Lambda$ which is a compact set, we can assume (without loss of generality) that $\lambda(x^k) \underset{k \in K}{\longrightarrow} \lambda^* \in \Lambda$. Then, by virtue of Lemma 5.1(a), $d_N^k \underset{k \in K}{\longrightarrow} d_N^*$. If we suppose, by a contradiction way, that x^* is not Pareto KKT-stationary, then according to Proposition (4.1)(ii)–(iii), d_N^* would be a multiobjective reduced descent direction of F at $x^* \in S$. Hence, according to the definition (8),

$$U_N(x^*) d_N^* < 0. \tag{16}$$

We can see from (15) that this would imply $t_k \underset{k \in K}{\longrightarrow} 0$. In particular, for any $p \in \mathbb{N}$, $t_k < \frac{1}{2^p}$ for any $k \in K$ sufficiently large. Recall that, by the implicit function theorem applied at x^*, there exists a unique mapping $\psi^* : W^* \longrightarrow V^*$, where W^* (resp. V^*) is a neighbourhood of x_N^* (resp. $x_B^* = \psi^*(x_N^*)$) such that $G\big(\psi^*(x_N), x_N\big) = 0$ for all $x_N \in W^*$. This applies to each $x_N^k + \frac{1}{2^p} d_N^k$ as $(k, p) \to +\infty$, since $x_N^k + \frac{1}{2^p} d_N^k \longrightarrow x_N^*$; hence, for all $p \in \mathbb{N}$ and $k \in K$, both sufficiently large,

$$G\Big(\psi^*(x_N^k + \frac{1}{2^p} d_N^k), x_N^k + \frac{1}{2^p} d_N^k\Big) = 0. \tag{17}$$

On the other hand, since x^* is assumed to be not Pareto KKT-stationary, then using the hypothesis $(\mathcal{H})$, x^* would be nondegenerate, so that $a_B < x_B^* = \psi^*(x_N^*) < b_B$. Thus, by the continuity of ψ^*, we would also have that, for all $p \in \mathbb{N}$ and $k \in K$, both sufficiently large,

$$a_B < \psi^*\Big(x_N^k + \frac{1}{2^p} d_N^k\Big) < b_B. \tag{18}$$

Now, using Lemma 5.1(b), we have that $0 < \frac{1}{2^p} \leq \bar{t}$ for all p large enough, and this shows, according to Proposition 3.1(ii), that for any $p \in \mathbb{N}$ sufficiently large and $k \in K$,

$$a_N \leq x_N^k + \frac{1}{2^p} d_N^k \leq b_N. \tag{19}$$

Subsequently, we would have, following (17)–(19), that for all $p \in \mathbb{N}$ and $k \in K$, both sufficiently large,

$$\Big(\psi^*\big(x_N^k + \frac{1}{2^p} d_N^k\big), x_N^k + \frac{1}{2^p} d_N^k\Big) \in S.$$

However, given the previous assertion that $t_k < \frac{1}{2^p}$, this would mean that, in Step 5 of the GRJ algorithm, for all $p \in \mathbb{N}$ and $k \in K$, both sufficiently large, the Armijo inequality would fail to hold; that is,

$$F\Big(\psi^*\big(x_N^k + \frac{1}{2^p} d_N^k\big), x_N^k + \frac{1}{2^p} d_N^k\Big) \not< F(x^k) + \beta \frac{1}{2^p} U_N(x^k) d_N^k. \tag{20}$$

Fixing $p \in \mathbb{N}$ sufficiently large and passing onto the limit when $k \nearrow +\infty$, we would obtain that

$$F\Big(\psi^*\big(x_N^* + \frac{1}{2^p} d_N^*\big), x_N^* + \frac{1}{2^p} d_N^*\Big) \not< F(x^*) + \beta \frac{1}{2^p} U_N(x^*) d_N^*.$$

Since $p \in \mathbb{N}$ was arbitrary large enough, this would be in contradiction with Proposition 3.1(iv), and this proves that x^* is well Pareto KKT-stationary for (MOP). □

6 Numerical experiments

6.1 Test problems and implementation

In this section, we present computational results obtained through the proposed GRJ method. We also provide comparisons with three other multicriteria methods: ZMO (a Zoutendijk-like method [31]), MOSQP (an SQP-type approach [22]) and the evolutionary NSGA-II method (the nondominated sorting genetic algorithm [10]). All codes were implemented on a machine equipped with 1.90 GHz Intel(R) Core(TM) i5 CPU and 16 Go memory. GRJ, ZMO, and MOSQP were coded in SCILAB-2024.1.0, while NSGA-II was executed using MATLAB's predefined function "gamultiobj" (MATLAB R2024b).

Thirty constrained multicriteria optimization test problems, selected from the literature, were considered in this experiment and are summarized in Table 1, where 'OV', 'L' and 'NL', respectively, indicate the number of original variables, the number of linear constraints and the number of nonlinear constraints. We selected the 'Disc Brake' and 'Welded Beam' problems, which are derived from real-world engineering design applications [35]. Detailed descriptions of these two models, along with further analysis of the results, are provided at the end of this section. Also, we introduced the problem 'EL3' below showing the importance of the numerical aspect of the tangent cone $\mathcal{T}_S(x)$, in the case where no feasible direction exists, i.e., $\mathcal{A}_S(x) = \emptyset$.

$$\begin{array}{lll} \text{(EL3)} & \text{Minimize} & \left(x_2^3 + \log\left(x_1^2 + 1\right), \sin\left(\dfrac{x_1}{x_2 + 2}\right)\right), \\ & \text{subject to} & x_1^2 + x_2^2 = 1, \\ & & 0 \leq x_i \leq 1, \quad i = 1, 2. \end{array} \tag{21}$$

Problem	Ref.	r	OV	L	NL	Problem	Ref.	r	OV	L	NL
ABC_comp	[26]	2	2	2	1	LAP1	[5]	2	2	1	2
BNH	[8]	2	2	0	2	LAP2	[5]	2	30	0	1
CF11	[44]	2	3	0	2	LIR-CMOP1	[19]	2	30	0	2
CPT1	[9]	2	2	0	30	LIR-CMOP2	[19]	2	30	0	2
Disc Brake	[35]	2	4	2	3	LIR-CMOP3	[19]	2	30	0	3
DTLZ0	[11]	3	3	0	2	liswetm	[27]	2	7	5	0
DTLZ9	[11]	3	30	0	2	MOLPg_001	[38]	3	8	8	0
EL3	(21)	2	2	0	1	MOLPg_002	[38]	3	12	13	0
ex001	[7]	2	5	1	2	MOLPg_003	[38]	3	10	12	0
ex002	[41]	2	5	0	4	OSY	[8]	2	6	4	2
ex003	[8]	2	2	0	2	SRN	[8]	2	2	1	1
GE1	[14]	2	2	0	1	Tamaki	[35]	3	3	0	1
GE4	[14]	3	3	0	1	TLK1	[39]	2	2	4	0
Hanne4	[6]	2	2	0	1	TNK	[8]	2	2	0	2
hs05x	[25]	3	5	6	0	Welded Beam	[35]	2	4	1	3

Table 1: Linearly and nonlinearly constrained multicriteria test problems

All these problems were solved starting with the same population of 200 selected individuals. To ensure that all compared solvers generate the same number of 200 final solutions, and since MATLAB's NSGA-II solver may provide fewer than 200, we set "PopulationSize" option to 571 for NSGA-II, while all other NSGA-II options were left with the default settings.

In our implementation, the three codes GRJ, ZMO and MOSQP use the value 0.25 as Armijo's constant, and

$$\min\left(\mathcal{P}_x\right) < 10^{-6}$$

as the stopping criterion, where $\mathcal{P}_x$ denotes the direction-finding subproblem specific to each of the three solvers. The latter was solved, for GRJ, by coding a standard RGM based on Wolfe's continuous scheme (see, e.g., [18] for details), while for ZMO and MOSQP, it was solved using SCILAB's predefined function "qld" for linear quadratic programming problems.

Step 5 of the GRJ algorithm, which involves computing feasible Armijo's steplengths, relies on the results presented in Proposition 3.1. This step is carried out using the Newton method, following the procedure outlined below:

Feasible Armijo line search pseudocode

Step 0: Initialization.

- Fix a tolerance $\varepsilon > 0$ (e.g., $\varepsilon = 10^{-6}$) and a maximum number of iterations $L \in \mathbb{N}$ (e.g., $L = 200$).

- Set $t = t_N$, $y_B^1 = x_B$ and $l = 1$.

Step 1: Newton rule.

- Set $y_B^{l+1} = y_B^l - A_B^{-1}\Big(y_B^l, x_N + td_N\Big)G\Big(y_B^l, x_N + td_N\Big)$.

Step 2: Feasible Armijo test.

- If

$$\left\|G\Big(y_B^{l+1}, x_N + td_N\Big)\right\| < \varepsilon, \quad a_B \leq y_B^{l+1} \leq b_B, \quad \text{and} \quad F\Big(y_B^{l+1}, x_N + td_N\Big) < F(x) + \beta t U_N d_N,$$

then STOP: the current implicit basic vector is set as $\psi\left(x_N + td_N\right) = y_B^{l+1}$.

Step 3: Update.

- If $l = L$, set $t = \dfrac{t}{2}$, $y_B^1 = x_B$ and $l = 1$; otherwise, set $l = l + 1$.

- Return to Step 1.

6.2 Performance measures and interpretation

To thoroughly analyse and compare the results obtained by the solvers, a set of performance metrics, as suggested in the literature (see, e.g., [2]), may be employed. These metrics essentially assess two key aspects: the convergence of the approximate solutions towards the Pareto front and their diversity. In this work, we have chosen three performance measures: *Purity* (P) [3], *Spread* (Δ^*) [43] and *Generational distance* (GD) [40]. These three measures use, as a proxy for the true Pareto front, the so-called *reference Pareto front*, defined as:

$$F_p = \left\{ y^* \in \bigcup_{s\in\mathcal{S}} F_{p,s} \;:\; \nexists y \in \bigcup_{s\in\mathcal{S}} F_{p,s},\; y < y^* \right\},$$

where $F_{p,s}$ stands for the approximated Pareto front of problem $p \in \mathcal{P}$ provided by method $s \in \mathcal{S}$; $\mathcal{P}$ denotes the set of tested problems and $\mathcal{S}$ the set of considered solvers.

- **Purity** (P). This metric consists in measuring the proportion of points in $F_{p,s}$ admitted in F_p:

  $$P_{p,s} = \frac{|F_{p,s} \cap F_p|}{|F_{p,s}|}.$$

  Clearly $P_{p,s} \in [0, 1]$ and the extreme values are significant in the sense that a value $P_{p,s}$ close to 1 indicates better performance, while a value equal to 0 implies that the solver is unable to generate any point of F_p.

- **Spread** (Δ^*). This metric measures the diversity and the dispersion of the points in $F_{p,s}$ with respect to F_p:

  $$\Delta^*_{p,s} = \frac{\sum_{j=1}^{r} d\left(y_j^*, F_{p,s}\right) + \sum_{y\in F_p} \left| d\left(y, F_{p,s} \setminus \{y\}\right) - \bar{d} \right|}{\sum_{j=1}^{r} d\left(y_j^*, F_{p,s}\right) + |F_p|\, \bar{d}},$$

  where $d(y, A) = \min_{a\in A} \|y - a\|$ is the Euclidean distance from the vector y to the set A; here, y_j^* denotes the minimum value of the criterion f_j in F_p, and $\bar{d}$ represents the average distance between each solution $y \in F_p$ and the set $F_{p,s} \setminus \{y\}$. Note that lower values of $\Delta^*_{p,s}$ reflects more uniform distributions of the generated solutions.

- **Generational Distance** (GD). Measuring the convergence, this metric represents how far $F_{p,s}$ is from F_p:

  $$GD_{p,s} = \frac{1}{|F_{p,s}|} \sqrt{\sum_{y\in F_{p,s}} d^2\left(y, F_p\right)}.$$

  Obviously, lower values of $GD_{p,s}$ are requested.

Table 2 lists the numerical results obtained by the four considered methods. The column 'CPU' indicates the average execution time (in seconds). The best scores are in bold. To analyse these results and look for significant differences between the compared solvers, the *performance profile* is used (see, e.g., [12]). Recall that a profile is a graphical representation of the (cumulative) distribution function ρ whose outputs are a solver's scores on the set of test problems against a performance metric. More precisely, given the measure $m_{p,s}$ (set here to 1/P, Δ^*, GD or CPU) by solver $s \in \mathcal{S}$ for solving $p \in \mathcal{P}$, the corresponding distribution function is given formally by

$$\rho_s = \frac{|\{p \in \mathcal{P} \;:\; r_{p,s} \le \alpha\}|}{|\mathcal{P}|},$$

where $r_{p,s} = m_{p,s}/ \min\{m_{p,s} : s \in S\}$. The purity measure $m_{p,s}$ is set to $1/P$ in order that all the considered metrics exhibit the same asymptotic behaviour, in a sense that the smaller the measure, the better the solver. Note that at the threshold $\alpha = 1$, $\rho_s(\alpha)$ gives us the largest number of problems among the best solved by s according to the analysed performance. However, a value $\rho_s(\alpha)$ attaining 1 means that all the problems $p \in \mathcal{P}$ have been solved by solver s at the threshold α. Thus, the best overall performance of a solver is that which reaches the value 1 for the smallest value of α.

Problems	GRJ				ZMO			
	CPU	P	$\Delta^\star$	GD	CPU	P	$\Delta^\star$	GD
ABC_comp	**0.015935**	0.99	**0.902345**	0.0000076	0.021454	**1**	0.953210	**0**
BNH	1.436124	0.915	0.810000	0.0000086	**0.004504**	**1**	0.995432	**0**
CF11	0.415047	0.885	0.920000	**0**	0.058855	0.61	**0.902345**	**0**
CPT1	1.712818	0.97	**0.765432**	0.000003	0.055902	0.98	0.895432	0.000003
Disc Brake	1.242267	0.69	**0.310123**	0.0031797	1.817904	0.035	0.860987	0.004630
DTLZ0	0.410519	**0.805**	0.705678	0.000615	**0.000827**	0	0.765432	0.001699
DTLZ9	**0.209932**	0.9	0.254321	**0**	1.004320	0.08	0.989765	0.000092
EL3	**0.005883**	**1**	**0.285432**	**0**	0.005889	**1**	0.925678	**0**
ex001	0.112501	**1**	0.415098	**0**	0.020554	0.33	**0.350987**	0.276242
ex002	0.027517	0.46	**0.785432**	0.0027654	**0.010278**	0	0.825678	0.042952
ex003	0.044351	0.535	0.811023	0.0003054	**0.038164**	**0.565**	0.985432	0.000682
GE1	0.018449	**0.995**	0.870987	**0**	**0.002708**	**0.995**	0.990987	**0**
GE4	0.048618	0.985	0.632098	0.0000041	0.037197	**0.99**	0.991098	0.000004
Hanne4	0.024853	0.56	**0.707098**	0.0004533	0.030180	0.46	0.998098	0.000614
hs05x	1.008168	0.975	0.980987	0.0006024	**0.006076**	0.01	0.970987	0.019143
LAP1	1.822010	0.755	**0.505432**	**0.000108**	0.028570	**0.965**	0.640987	0.000128
LAP2	0.087737	0.87	**0.180123**	0.0058377	0.082840	**1**	0.995432	**0**
LIR-CMOP1	0.389043	**0.837**	0.840987	0.000219	**0.035736**	0	0.965098	0.003135
LIR-CMOP2	**0.135677**	**0.9**	**0.750987**	0.000038	0.135736	0.02	0.997453	0.003135
LIR-CMOP3	0.031187	**0.995**	**0.509871**	**0**	**0.010200**	0.62	0.740987	0.000065
liswetm	0.154443	**0.61**	**0.150987**	**0.000048**	**0.015055**	0	0.930987	0.000815
MOLPg_001	**0.184877**	0.91	0.955432	0.000004	0.550163	0.81	0.980987	0.000005
MOLPg_002	1.200010	0.625	0.740987	0.000758	**0.015298**	0	0.720987	0.040700
MOLPg_003	1.005378	**0.99**	0.700987	**0**	0.042475	0	0.560987	0.021736
OSY	1.539581	0.47	0.923444	0.003442	**0.061214**	0	**0.781341**	0.019456
SRN	1.937231	**0.945**	0.860987	**0**	**0.010315**	0.9	0.940987	0.006855
Tamaki	1.541475	0.83	0.700987	**0**	**0.002469**	**1**	**0.530987**	**0**
TLK1	**0.013301**	0.42	**0.840987**	0.000101	0.033947	0.39	0.950097	0.000080
TNK	0.459437	**0.395**	0.830987	0.0003194	**0.013540**	0.275	**0.760987**	0.004407
Welded Beam	1.137963	0.91	**0.740987**	0.0033562	13.20645	0	0.960987	1.226983

Problems	MOSQP				NSGA-II			
	CPU	P	$\Delta^\star$	GD	CPU	P	$\Delta^\star$	GD
ABC_comp	0.020315	0.99	0.980987	0.000017	0.857500	0.41	0.970000	0.000906
BNH	0.163576	0.105	0.992109	0.010358	0.013100	0.84	**0.780000**	0.001654
CF11	**0.011990**	0.59	1.000000	**0**	0.021800	**1**	0.994567	**0**
CPT1	**0.017768**	0.74	0.900000	0.000025	0.191800	**1**	1.000000	**0**
Disc Brake	0.021562	0	0.360000	0.006729	**0.0084**	**0.84**	0.840012	**0.001951**
DTLZ0	0.014875	0.495	0.960000	**0.000528**	0.0094	0.485	**0.610000**	0.003421
DTLZ9	0.395620	**0.93**	0.840000	0.005980	0.3825431	0.08	**0.070981**	0.000092
EL3	0.012636	0	0.659765	0.010208	0.0143	0.99	0.591220	0.000028
ex001	**0.087176**	0	1.000000	1.875443	4.9669	0	0.991197	**0**
ex002	0.029159	**0.55**	0.967787	**0.001515**	0.038400	0.085	1.000000	0.021188
ex003	0.065472	0.395	**0.793245**	0.000364	0.147100	0	0.985023	**0.000061**
GE1	0.031198	**0.995**	1.000000	**0**	0.010200	0.62	**0.760000**	0.000040
GE4	**0.019889**	0.97	0.992501	**0**	0.01200	0.87	**0.570000**	0.000458
Hanne4	0.022001	0.355	0.980000	**0.000308**	**0.014400**	**0.775**	0.900000	0.000643
hs05x	0.056837	**1**	0.990000	**0**	3.059300	0.941	**0.905401**	0.010439
LAP1	0.023713	0.255	0.970350	0.000203	**0.010900**	0.78	0.870000	0.000176
LAP2	0.024061	0.27	0.790000	0.006000	**0.016500**	0.075	0.960000	0.059720
LIR-CMOP1	0.059540	0.8	**0.806701**	0.000053	0.1824925	0.22	1.000000	**0.000002**
LIR-CMOP2	0.236534	0.53	0.950276	**0.000022**	2.1357355	0.1	0.993204	0.041562
LIR-CMOP3	0.323865	0.76	0.805467	0.000376	**0.010200**	0.62	0.760000	0.000065
liswetm	0.016368	0.385	1.000000	0.000410	1.119700	0.515	0.830000	0.000244
MOLPg_001	0.1888712	**0.96**	**0.934124**	0.000234	0.210634	0.9	0.993204	**0**
MOLPg_002	0.017967	**0.68**	0.995119	**0**	1.95200	0.075	**0.703005**	0.075590
MOLPg_003	**0.032949**	0.79	0.968578	0.001239	1.693900	0.19	**0.503204**	0.042700
OSY	1.582346	**0.525**	0.998123	**0.001645**	1.837200	0.25	0.934551	0.001930
SRN	0.036814	**0.945**	0.967634	**0**	2.393800	0.685	**0.773497**	0.004229
Tamaki	0.02418	0.9	0.843457	**0**	0.017700	0.99	0.952078	0.000007
TLK1	0.019948	0.225	0.996780	0.000042	0.959700	**0.985**	0.990840	**0.000004**
TNK	1.014764	0.09	1.000000	**0.000245**	0.97000	0.33	1.000000	0.003962
Welded Beam	**0.024213**	0.05	0.999732	0.178934	0.0875	**0.97**	0.950000	**0.000621**

Table 2: Multiobjective performance measurements

As illustrated in Fig. 1, the GRJ method consistently demonstrates good performance with respect to the P metric, compared to the other methods. Indeed, the best value $\rho(\alpha) = 1$ is achieved by GRJ at a relatively small threshold α, whereas the other methods fail to reach this value. This observation is further supported by the results in Tab. 2, where the purity values reported for the ZMO, MOSQP and NSGA-II methods are in some cases very close or equal to zero. This suggests that these methods fail to reach the reference Pareto front for certain problems.

In terms of the Δ^* and GD metrics, the four methods appear to be generally comparable. Although NSGA-II is widely recognized for its effectiveness in spread, the observed profile indicates good overall performance across all solvers, with a slight advantage noted for GRJ and NSGA-II. Similar observations can be made regarding the GD metric, noting this time a slight advantage of the three methods over NSGA-II, which is well known for its limited convergence toward the Pareto front.

Regarding CPU time, as shown in Fig. 1 and Tab. 2, the three methods exhibit a slight speed advantage over GRJ. This can be attributed to the inherent characteristics of the GRJ algorithm and its operational mechanism. More specifically, the algorithm's iterative process and heuristic strategies, such as the basis selection procedure in Step 1 and the line search procedure in Step 5 (as previously described), can be time-consuming. Nevertheless, these components enable efficient exploration of the solution space, which helps to explain its performance differences relative to the other methods. The graphical representation of the approximated Pareto fronts illustrated in Fig. 2–3 also supports these interpretations. Note also that ZMO generally appears to be faster than MOSQP based on the scores given in Tab. 2, except for the case where ZMO relatively took a long time to solve the 'Welded Beam' design problem. This may explain the influence of MOSQP's CPU performance profile on ZMO observed in Fig. 1. However, as reported in Tab. 2, the execution time remains generally very small and reasonable for all the considered solvers.

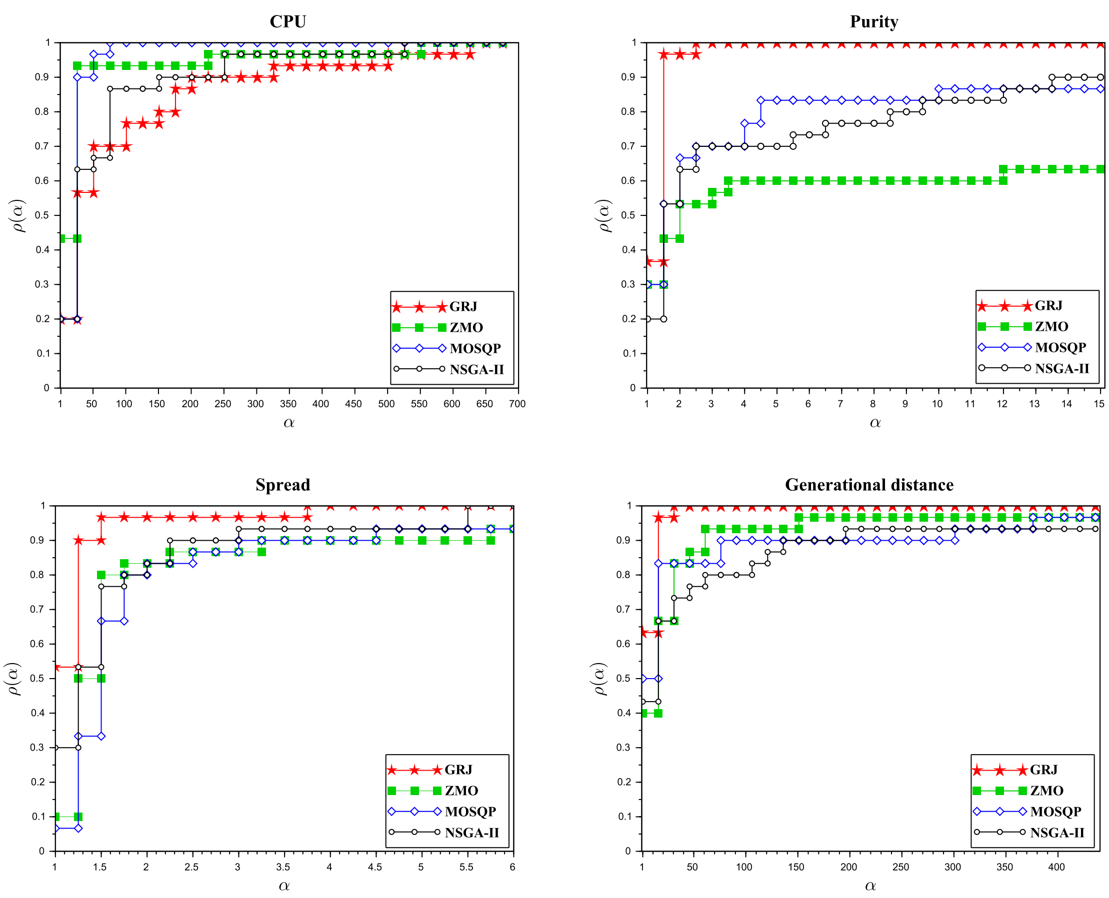

Figure 1: Performance profiles

6.3 Real-world applications

As mentioned earlier, we conclude this section by providing additional details on the two real-world engineering problems: 'Disc Brake' and 'Welded Beam'.

6.3.1 'Disc Brake' design problem

This problem consists to minimize simultaneously the mass of the brake and the stopping time. The variables represent the inner radius of the disc, the outer radius, the engaging force, and the number of friction surfaces. The constraints for the design include a minimum distance between the inner and outer radii, a maximum allowable brake length, and limitations related to pressure, temperature, and torque. The problem is a nonlinearly constrained bicriteria problem (see [35] for further details), formally defined as follows:

$$
\begin{aligned}
\text{(Disc Brake)} \qquad \text{Minimize} \quad & \left(4.9 \times 10^{-5}\left(x_2^2 - x_1^2\right)(x_4 - 1),\, \frac{9.82 \times 10^6 \left(x_2^2 - x_1^2\right)}{x_3 x_4 \left(x_2^3 - x_1^3\right)}\right), \\
\text{subject to} \quad & (x_2 - x_1) + 20 \leq 0, \quad 2.5\,(x_4 + 1) - 30 \leq 0, \\
& \frac{x_3}{3.14\left(x_2^2 - x_1^2\right)} - 0.4 \leq 0, \quad \frac{2.22 x_3 \left(x_1^3 - x_2^3\right)}{10^3 \times \left(x_2^2 - x_1^2\right)^2} - 1 \leq 0, \\
& \frac{2.66 x_3 x_4 \left(x_1^3 - x_2^3\right)}{10^2 \left(x_2^2 - x_1^2\right)^2} + 900 \leq 0, \\
& 55 \leq x_1 \leq 80, \quad 75 \leq x_2 \leq 110, \\
& 10^3 \leq x_3 \leq 3 \times 10^3, \quad 2 \leq x_4 \leq 20.
\end{aligned}
$$

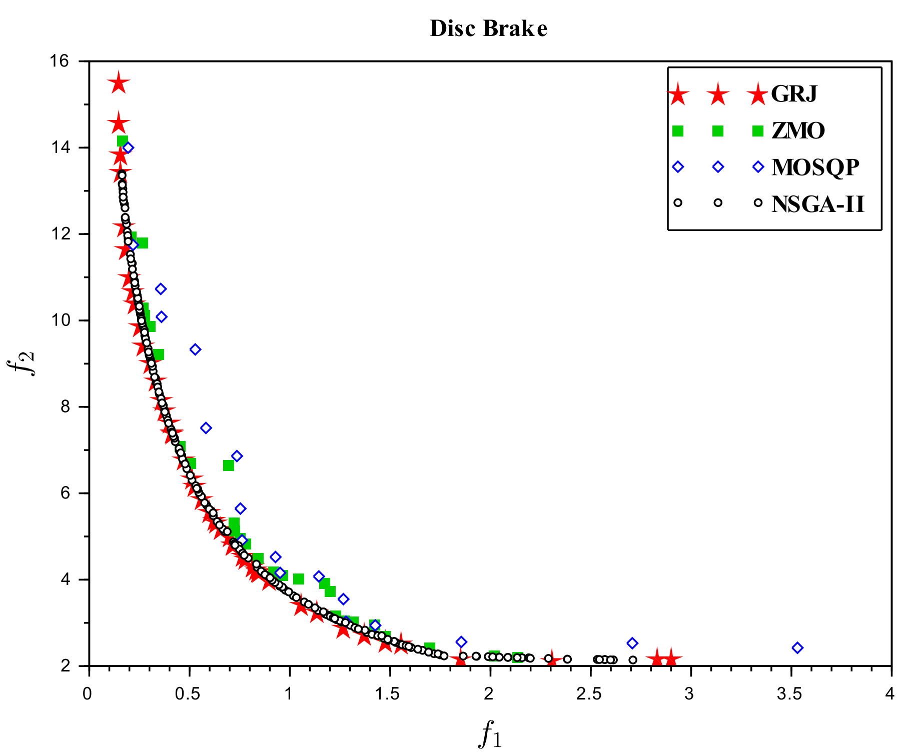

Figure 2: Best Pareto front approximations for the 'Disc Brake' design problem by GRJ, ZMO, MOSQP and NSGA-II

The graphical representation of the approximate Pareto fronts in Fig. 2 obtained by the four methods highlights the superior ability of GRJ and NSGA-II to better explore the Pareto front for the ‘Disc Brake’ problem, in contrast to ZMO and MOSQP, which face serious challenges. Nevertheless, it is important to note that MOSQP slightly outperforms in this example ZMO, as it manages to find some relatively interesting solutions.

6.3.2 ‘Welded Beam’ design problem

This problem involves minimizing both the cost of fabrication and the deflection of the beam under an applied load subject to some constraints. The two objectives are inherently conflicting, since reducing deflection generally leads to higher manufacturing costs, primarily due to setup, material, and welding labor costs. The design involves four decision variables and four nonlinear constraints: shear stress, normal stress, weld length, and buckling limitations (see [35] for more details). Its formal definition is given as follows:

$$
\text{(Welded Beam)} \qquad \text{Minimize} \quad \left(1.10471x_1^2x_2 + 0.04811x_3x_4(14+x_3), \frac{2.1952}{x_3^3x_4}\right),
$$

$$
\text{subject to} \quad \sqrt{\tau'(x)^2 + \tau''(x)^2 + \frac{x_2\tau'(x)\tau''(x)}{\sqrt{0.25\big(x_2^2+(x_1+x_3)^2\big)}}} \leq 13600,
$$

$$
\frac{1}{x_3^3x_4} \leq \frac{3}{504}, \;\; x_3x_4^3(0.0282346x_3 - 1) \leq \frac{6\times10^3}{64746.022},
$$

$$
x_1 \leq x_4, \;\; 0.125 \leq x_1, x_4 \leq 5, \quad 0.1 \leq x_2, x_3 \leq 10,
$$

where $\tau'(x) = \frac{6\times10^3}{\sqrt{2}x_1x_2}$ and $\tau''(x) = \frac{3\times10^3\left(14+0.5x_2\sqrt{0.25(x_2^2+(x_1+x_3)^2)}\right)}{0.707x_1x_2\left(\frac{x_2^2}{12}+0.25(x_1+x_3)^2\right)}$.

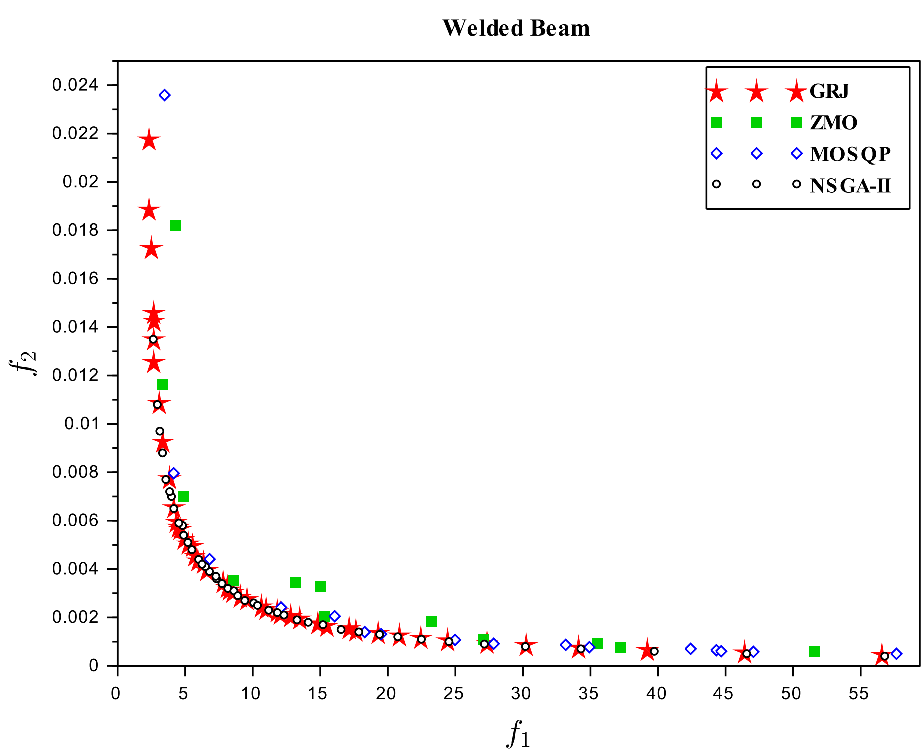

Figure 3: Best Pareto front approximations for the ‘Welded Beam’ design problem by GRJ, ZMO, MOSQP and NSGA-II

In the context of the 'Welded Beam' problem, as shown in Fig. 3, all methods succeed in approximating significant regions of the Pareto front. However, the resulting dispersions differ from one method to another, so that none can be objectively favoured in this regard. In terms of convergence, it is also observed that the ZMO method is dominated by the other methods, which remain in strong competition with each other. This clearly demonstrates the effectiveness of our GRJ method in solving real-world problems. Moreover, the consistent performance of GRJ across various metrics highlights its robustness and reliability as a practical optimization approach.

Statements and Declarations. The authors have no pertinent declarations concerning conflicts of interest, financial or non-financial interests, competing interests or other statements to disclose.

References

[1] Abadie, J., Carpentier, J.: Generalization of the Wolfe reduced gradient method to the case of nonlinear constraintes. In: Optimization (R. Fletcher, ed.) Academic Press, London and New York, 37–47 (1969)

[2] Audet, C., Bigeon, J., Cartier, D., et al.: Performance indicators in multiobjective optimization. Eur. J. Oper. Res. 292, 397–422 (2021)

[3] Bandyopadhyay, S., Pal, S.K., Aruna, B.: Multiobjective GAs, quantitative indices, and pattern classification. IEEE Trans . Syst. Man Cyber. B Cyber. 34, 2088–2099 (2004)

[4] Bazaraa, M.S., Sherali, H.D., Shetty, C.M.: Nonlinear Programming: Theory and Algorithms. John Wiley and Sons, New York (2006)

[5] Cocchi, G., Lapucci, M.: An augmented Lagrangian algorithm for multi-objective optimization. Computat. Optim. Appli. 77, 29–56 (2020)

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

[7] Das, I., Dennis, J.E.: A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. Structural Optimization 14, 63–69 (1997). https://doi.org/10.1007/BF01197559

[8] Deb, K.: Multi-objective genetic algorithms: problem difficulties and construction of test problems. Evolutionary Computation 7, 205–230 (1999)

[9] Deb, K., Pratap, A., Meyarivan, T.: Constrained Test Problems for Multi-objective Evolutionary Optimization. In: Zitzler E et al (ed) Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO-01), 284–298 (2001)

[10] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. 6, 182–197 (2002)

[11] Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable Test Problems for Evolutionary Multiobjective Optimization. Springer, London (2005)

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

[13] Drummond, L.M.G., Iusem, A.N.: A projected gradient method for vector optimization problems. Computat. Optim. Appli. 28, 5–29 (2004)

[14] Eichfelder, G.: An adaptive scalarization method in multiobjective optimization. SIAM J. Optim., 19(4), 1694–1718 (2009)

[15] El Maghri, M.: A free reduced gradient scheme. Asian Journal of Mathematics and Computer Research 9, 228–239 (2016)

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

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

[18] El Maghri, M., Elboulqe, Y.: A reduced Jacobian method with full convergence property. Optimization Letters 18, 1647–1671 (2024)

[19] Fan, Z., Li, W., Cai, X. et al.: An improved epsilon constraint-handling method in MOEA/D for CMOPs with large infeasible regions. Soft Comput. 23, 12491–12510 (2019). https://doi.org/10.1007/s00500-019-03794-x

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

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

[22] Fliege, J., Vaz, A.I.F.: A method for constrained multiobjective optimization based on SQP techniques. SIAM J. Optim. 24, 2091–2119 (2016)

[23] García-Palomares, U.M., Burguillo-Rial, J.C., González-Castaño, F.J.: Explicit gradient information in multiobjective optimization. Operation Research Letters 36, 722–725 (2008)

[24] Goh, C.J., Yang, X.Q.: Duality in Optimization and Variational Inequalities. Taylor and Francis, London (2002)

[25] Hock, W., Schittkowski, K.: Test Examples for Nonlinear Programming Codes. Springer–Verlag, Berlin, (1981)

[26] Hwang, C.L., Masud, A.S.M.: Multiple Objective Decision Making Methods and Applications: A State-of-the-Art Survey. Springer Science & Business Media, Berlin (1979)

[27] Leyffer, S.: A note on multiobjective optimization and complementarity constraints. Preprint ANL/MCS-P1290-0905, Mathematics and Computer Science Division, Argonne National Laboratory, Illinois (2005)

[28] Mangasarian, O.L.: Nonlinear Programming. SIAM, Philadelphia (1994)

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

[30] Mokhtar-Kharroubi, H.: Sur la convergence théorique de la méthode du gradient réduit géneralisé. Numerische Mathematik 31, 13–35 (1980)

[31] Morovati, V., Pourkarimi, L.: Extension of Zoutendijk method for solving constrained multiobjective optimization problems. Eur. J. Oper. Res. 273, 44–57 (2019)

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

[33] Pérez, L.R.L, Prudente, L.F.: Nonlinear conjugate gradient methods for vector optimization. SIAM Journal on Optimization 28, 2690–2720 (2018)

[34] Qu S., Goh, M., Chand F.T.S.: Quasi-Newton methods for solving multiobjective optimization. Operation Research Letters 39, 397–399 (2011)

[35] Ray, T., Liew, K. M.: A swarm metaphor for multiobjective design optimization. Engineering Optimization, 34, 141–153 (2002)

[36] Schittkowski K.: On the convergence of a generalized reduced gradient algorithm for nonlinear programming problems. Optimization 17, 731–755 (1986)

[37] Smeers, Y.: Generalized reduced gradient method as an extension of feasible direction methods. Journal of Optimization Theory and Applications 22, 209–226 (1977)

[38] Steuer, R.E.: Multiple Criteria Optimization: Theory, Computation, and Application. Wiley Series in Probability and Mathematical Statistics (1986)

[39] Thang T.N., Luc D.T., Kim N.T.B.: Solving generalized convex multiobjective programming problems by a normal direction method. Optimization 65, 2269–2292 (2016)

[40] Van Veldhuizen, D.A., Lamont, G.B.: On measuring multiobjective evolutionary algorithm performance. Proc. 2000 IEEE Cong. Evol. Comput. (2000). https://doi.org/10.1109/CEC.2000.870296

[41] Wang, J.-F., Renaud, J. E.: Automatic differentiation in multi-objective collaborative optimization. In Proceedings of the 3rd WCSMO, Buffalo, New York, May (1999)

[42] Wolfe, P.: Methods of nonlinear programming. In: Graves, R.L., Wolfe, P. (Eds.) Recent Advances in Mathematical Programming, pp. 67–86. McGraw-Hill, New York (1963)

[43] Zhou, A., Jin, Y., Zhang, Q., Sendhoff, B., Tsang, E.: Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion. In IEEE international conference on evolutionary computation, 892–899 (2006) https://doi.org/10.1109/CEC.2006.1688406

[44] Zhou, Y., Xiang, Y., He, X.: Constrained multiobjective optimization: Test problem construction and performance evaluations. IEEE Transactions on Evolutionary Computation, 25, 172–186 (2021)