Journal of Optimization Theory and Applications (2018) 179:917–943
https://doi.org/10.1007/s10957-018-1362-x

Reduced Jacobian Method

Mounir El Maghri[1] · Youssef Elboulqe[1]

Received: 14 March 2018 / Accepted: 26 July 2018 / Published online: 10 August 2018

Abstract

In this paper, we present the Wolfe's reduced gradient method for multiobjective (multicriteria) optimization. We precisely deal with the problem of minimizing nonlinear objectives under linear constraints and propose a reduced Jacobian method, namely a reduced gradient-like method that does not scalarize those programs. As long as there are nondominated solutions, the principle is to determine a direction that decreases all goals at the same time to achieve one of them. Following the reduction strategy, only a reduced search direction is to be found. We show that this latter can be obtained by solving a simple differentiable and convex program at each iteration. Moreover, this method is conceived to recover both the discontinuous and continuous schemes of Wolfe for the single-objective programs. The resulting algorithm is proved to be (globally) convergent to a Pareto KKT-stationary (Pareto critical) point under classical hypotheses and a multiobjective Armijo line search condition. Finally, experiment results over test problems show a net performance of the proposed algorithm and its superiority against a classical scalarization approach, both in the quality of the approximated Pareto front and in the computational effort.

Keywords Multiobjective optimization · Nonlinear programming · Pareto optima · KKT-stationarity · Descent direction · Reduced gradient method

Mathematics Subject Classification 90C29 · 90C30 · 90C52

1 Introduction

Nowadays, in many areas of applications arising in decision-making problems, for instance in economics, management science, engineering, medicine (the literature on

✉ Mounir El Maghri
elmaghri@yahoo.com

Youssef Elboulqe
elboulqeyoussef@gmail.com

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

this subject is abundant, e.g., [1]), practitioners are often confronted to optimization problems, which by nature are multiobjective. Such problems minimize several objectives simultaneously, and they cannot be described adequately by only one-objective function. For example, optimal single-objective solutions are often pursued at the expense of the much broader applicability of designs and solutions that satisfy multiple objectives.

There is often a conflict between the objectives when a convex cone identifies a partial order, such as the natural ordering cone (nonnegative orthant). In general, there is no optimal solution minimizing the competing objective functions all at once. So the Pareto optima, also called efficient solutions, have to be considered. Taking account the principle, these solutions are incomparable and no single point can represent any of all the others. The problem becomes then to determine the set of Pareto points, or its image via the objectives, called the Pareto front. Since the latter is not necessarily a singleton, and is even generally infinite or unbounded, the best way to achieve this is to provide a good process giving rise to good approximations to the true Pareto front, so that decision makers can rank different options, depending on their preferences or their utilities.

Many computational methods were proposed for this task. Most employed approaches are based on scalarization techniques allowing to adapt the scalar optimization methods (see, e.g., [1–5]). It is now commonly known that using scalarization for a computational purpose may have some disadvantages. We can summarize the main difficulties, known on this approach, through the choice of the vector of weights (scalarization parameters), which is of capital importance. For example, it can happen, even in the convex case, that a dense subset of the weights corresponds to a discrete subset of the Pareto front, and vice versa. Moreover, the vector problem may have nice properties, while the scalarized ones may lose those qualities, for instance may be unbounded (see, e.g., [6]). Nevertheless, more sophisticated techniques, called adaptive scalarization methods, have been recently proposed in [7,8]. They propose strategies that try to control and adjust the current weights on the basis of antecedent scalar programs already solved. This novel vision could open future avenues of research for a new generation of scalarizing methods.

The first appearance of nonscalarizing methods was in the last decade, due to Fliege and Svaiter [9], who have suitably extended the Cauchy's steepest descent method as well as the Zoutendijk's feasible direction method. Next, the projected gradient method was extended by Graña Drummond and Iusem [10] and Newton's method by Fliege et al. [6]. Later on, the same line was pursued by García-Palomares et al. [11], proposing another way for computing the multiobjective search direction. Note that the multiobjective proximal methods, suggested by Bonnel et al. [12], can now be directly applied with this new generation of multiobjective descent methods.

The Wolfe's reduced gradient method (RGM) [13], one of fundamental and practical approaches in scalar optimization, combines techniques of both linear and nonlinear programming. As for Dantzig's simplex principle, the original problem is reduced to an unconstrained nonlinear program depending only on the so-called nonbasic variables, so that unconstrained optimization tools may be used. Since the beginning, this reduction strategy has received a particular attention from specialists and so, several important improvements were highlighted in theory as well as in programming (for more details, see the books [14,15] and/or the new contribution by El Maghri [16], for example).

In this paper, RGM is once again revisited, but now within the multiple objective framework for which up to now, the method knew no attempt. We propose to solve linearly constrained multiobjective nonlinear optimization problems by a direct extension of RGM, without scalarizing those programs. Our approach, called reduced Jacobian method (RJM), uses a special combination of reduced gradients for the finding-direction problem. Moreover, this problem is defined only in the image space (i.e., the space, where the images of the objective functions run) and has nice properties making it very suitable for its resolution. For example, it is convex, differentiable and has only one linear constraint. Solving this problem at a given iteration of the algorithm provides at the current point a reduced search direction common to all the objectives. The next iterate is obtained thanks to a multiobjective Armijo-type steplength along the descent direction. Under standard RGM hypotheses, the resulting algorithm is globally convergent to a Pareto KKT-stationary point, or the so-called Pareto critical point. Here, "globally convergent" means that starting from any feasible point, all sequences produced by the algorithm have decreasing objective function values and that all accumulation points of these sequences are critical points (see, e.g., the book [15]).

2 Multiobjective Descent Principle

Let $S = \{x \in \mathbb{R}^n \ : \ Ax = b,\ x \geq 0\}$, where $A \in \mathbb{R}^{m\times n}$ is a matrix of full rank $m < n$, $b \in \mathbb{R}^m$, and, let $F : x \in S \subset \mathbb{R}^n \longmapsto \mathbb{R}^r \ni F(x) = \big(f_1(x), \ldots, f_r(x)\big)$ be a $\mathcal{C}^1$ (continuously differentiable) vector-valued function on S, with $r \geq 1$.

We are dealing with the constrained nonlinear multiobjective program:[1]

$$\text{(MOP)} \qquad \text{Min } F(x) \quad \text{subject to} \quad Ax = b,\ x \geq 0.$$

The problem (MOP) "minimizes" simultaneously the r objective functions $f_i : x \longmapsto f_i(x)$, for all $i \in \{1, \ldots, r\}$, over S, in the following senses:

A solution $x^* \in S$ is said to be

- *weakly efficient* for (MOP), if $\nexists x \in S,\ F(x) < F(x^*)$;
- *efficient* (*Pareto optimum*) for (MOP), if $\nexists x \in S,\ F(x) \lneq F(x^*)$;
- *properly efficient* for (MOP), if $\exists C \subsetneq \mathbb{R}^r$ a pointed[2] convex cone satisfying $\mathbb{R}^r_+\backslash\{0\} \subseteq \text{int}C$, such that x^* is efficient for (MOP) with respect to the order $\leq_C$ induced by C.

We denote, respectively, by E_p, E_e and E_w, the sets of properly efficient points, efficient points and weakly efficient points. Taking into account the trivial inclusions

[1] For $y, y' \in \mathbb{R}^r$, $y < y'$ (resp. $y \leq y'$) means $y_i < y'_i$ (resp. $y_i \leq y'_i$) for all $i \in \{1, \ldots, r\}$; obviously, $y \lneq y'$ means $y \leq y'$ and $y \neq y'$.

[2] Pointed means $C \cap -C = \{0\}$, int$C$ stands for the topological interior of C, and, $y \leq_C y'$ means $y - y' \in -C$.

$E_p \subseteq E_e \subseteq E_w$, the determination of the whole weakly efficient set will include efficient and properly efficient ones.

The concept of *local* efficiency is defined by replacing S by $S \cap \mathcal{N}(x^*)$ in the above definitions, where $\mathcal{N}(x^*)$ is some neighborhood of x^*. Recall that, under some convexity assumptions, every local solution of the (MOP) is a global one. The image set of the efficient solutions is called *Pareto front*.

A multiobjective search direction is defined as follows: $d \in \mathbb{R}^n$ 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 *descent direction* of F at $x \in S$, if $\exists t_d > 0,\ \forall t \in \,]0, t_d\,],\ F(x + td) < F(x)$.

A sufficient condition for definition (b) is that (see [9])

$$JF(x)d < 0, \tag{1}$$

where $JF(x) = \left(\partial f_j(x)/\partial x_i\right)_{j,i}$ is the *Jacobian* $(r \times n)$-matrix of F at x. The jth line of $JF(x)$ is of course the gradient $\nabla f_j(x)$ of the jth objective f_j at x.

A necessary condition for a point $x^* \in S$ to be locally weakly efficient solution (in particular, efficient or properly efficient solution) for (MOP) is that

$$\nexists d \in \mathbb{R}^n \ \text{ a feasible descent direction of } F \text{ at } \ x^*. \tag{2}$$

Indeed, suppose the contrary. Then, by the very definition, there will exist $t^* > 0$ such that for all $t \in \,]0, t^*\,]$, $x^* + td \in S$ and $F(x^* + td) < F(x^*)$. Hence, for any $\varepsilon > 0$, we would have for all $t \in \,]0, \min\left(t^*, \frac{\varepsilon}{\|d\|}\right)[$, $x^* + td \in S \cap B(x^*, \varepsilon)$ and $F(x^* + td) < F(x^*)$, where $B(x^*, \varepsilon)$ is the open ball centered in x^* with radius ε. But this would contradict the hypothesis.

Definition 2.1 The vector mapping $F : S \subset \mathbb{R}^n \longrightarrow \mathbb{R}^r$ is said to be

(a) *convex* on S, if

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

(b) *strictly convex* on S, if

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

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

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

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

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

Remark 2.1 1. The pseudoconvexity (resp. strict pseudoconvexity) concept was introduced for differentiable vector (resp. scalar) functions in [17] (resp. in [18]). It is immediate that strict pseudoconvexity implies pseudoconvexity.

2. Recall that F is convex, iff F is componentwise convex. On the other hand, if F is componentwise pseudoconvex, then F is pseudoconvex (see [17, Theorem 9.2.3]), but not vice versa (see [19, Remark 5.3]). It is easy to verify that the same properties hold, respectively, for the strict convexity and the strict pseudoconvexity.
3. As in the scalar case, pseudoconvexity (resp. strict pseudoconvexity) is a weaker property than convexity (resp. strict convexity); the reciprocal being naturally false (see example *Test 2* in Sect. 5). This is a direct consequence of the fact that the graph of a convex (resp. strictly convex) differentiable functional is always above (resp. strictly above) its tangents.

Explicit necessary and sufficient efficiency conditions for our (MOP) are exhibited below and will be needed in the sequel.

Theorem 2.1 *A necessary condition for $x^* \in S$ to be locally weakly efficient solution for (MOP) is that one of the following two equivalent conditions is satisfied:*

(i) *There is no feasible descent direction of F at $x^* \in S$ satisfying (1), i.e.,*

$$\nexists d \in \mathbb{R}^n, \qquad JF(x^*)d < 0, \quad Ad = 0, \quad d_i \geq 0, \quad \forall i \in I(x^*),$$

where $I(x^*) = \big\{i \in \{1, \ldots, n\} : \;\; x_i^* = 0\big\}$ is the set of active variables.

(ii) *The point $x^* \in S$ satisfies the Pareto KKT-stationarity system, i.e.,*

$$\exists(\lambda^*, u^*, v^*) \in \mathbb{R}_+^r \backslash \{0\} \times \mathbb{R}_+^n \times \mathbb{R}^m, \;\; JF(x^*)^{\mathrm{T}}\lambda^* + A^{\mathrm{T}}v^* - u^* = 0, \;\; u^* \cdot x^* = 0,$$

where the symbol " $\cdot$ " stands for the scalar product.

These conditions become sufficient for the point x^* to be weakly efficient (resp. efficient) for (MOP), if F is pseudoconvex (resp. strictly pseudoconvex) at x^*.

A feasible point satisfying the system in Theorem 2.1(ii) is called *Pareto KKT-stationary* or *Pareto critical* for (MOP), and (λ^*, u^*, v^*) are the Lagrange multipliers. Note that Theorem 2.1(i) is easily obtained by contradiction with (2). The equivalence between (i) and (ii) is a direct application of Motzkin's alternative theorem (see, e.g., [20, p. 28]). The sufficiency is straightforwardly obtained by contraposition with (i), using the pseudoconvexity definitions. Another proof of sufficiency in an unconstrained Riemannian context can be found in [21]. The readers interested in necessary Pareto optimality conditions in a more general setting (spaces, ordering cone, constraints and/or objectives) may consult the paper by Durea et al. [22].

Remark 2.2 It is well known that a *Pareto KKT-stationary* point for a convex (MOP), with a multiplier $\lambda^* \in \mathrm{int}\mathbb{R}_+^r$, is a properly efficient solution to the (MOP) (see, e.g., [3]).

It is clear from Theorem 2.1 that if a point is not Pareto KKT-stationary, then it exists always at this point a multiobjective feasible descent direction. By consequent, as in scalar case, the general multiobjective descent scheme will conform to the following procedure:

❶ If x is Pareto KKT-stationary, then STOP;
❷ else, compute a multiobjective feasible descent direction d at x;
❸ choose a suitable steplength t of moving along the direction d;
❹ repeat the procedure with $x := x + td$.

Two central questions remain to be investigated: how to compute the search directions and how to check the stopping criterion. To this purpose, we shall proceed by extending the reduced gradient strategy to what we refer to as the *reduced Jacobian method* (RJM), which is the subject of the next section.

3 RJM Strategy

3.1 Reduced Jacobian

Since it is assumed that rank $A = m$, there exists a subset $B \subset \{1, \ldots, n\}$ such that A_B is an invertible submatrix with the partition $A = [A_B \; A_N]$ (rearranging the columns if necessary), where obviously $N = \{1, \ldots, n\} \backslash B$. Usually, B is called a *basis* of the matrix A. Rearranging also the so-called *basic variables* $x_B = (x_i)_{i \in B}$ and *nonbasic variables* $x_N = (x_i)_{i \in N}$, we can write $x = (x_B, x_N)$, so that

$$x \in S \Longleftrightarrow x = \left(A_B^{-1} b - A_B^{-1} A_N x_N, x_N\right) \geq 0. \tag{3}$$

A feasible solution x is called *nondegenerate*, if there exists a basis B such that $x_B > 0$; and it is said to be *degenerate*, if for any basis B, there exists $i \in B$ such that $x_i = 0$. The set S is in turn called nondegenerate, if any $x \in S$ is nondegenerate. Recall that a vertex $x \in S$ is characterized by (3) with $x_N = 0$, so its nondegeneracy means that $A_B^{-1} b > 0$. The classical degeneracy phenomenon, well known for the vertices of the standard polyhedron S, is a source of difficulties for both theory and computation in linear programming. Its generalization by Huard [23] to nonlinear programming is revealed to be necessary for any feasible point. The assumption holds particularly, if any m columns of the matrix A are linearly independent, and any vertex of S is nondegenerate, in which case, any feasible solution will have at most $n - m$ zero variables (see, e.g., [14, p. 603] or [15, p. 382]).

The Jacobian matrix $JF(x) = [JF_B(x) \; JF_N(x)]$ and the vector $d = (d_B, d_N)$ are partitioned in the same way; $JF_B(x)$ (resp. $JF_N(x)$) is of course the Jacobian matrix of F with respect to the basic variables (resp. nonbasic variables). Given a vector d_N, we define $d_B = -A_B^{-1} A_N d_N$ in such a way that $Ad = 0$ and

$$JF(x)d = U_N(x)d_N, \tag{4}$$

where $U_N(x) := JF_N(x) - JF_B(x)A_B^{-1}A_N$ is an $r \times (n - m)$ matrix that we call the *reduced Jacobian* matrix of F at x. Note that the jth row of $U_N(x)$ is the well-known *reduced gradient* of the objective f_j at x that we shall denote by $u_N^j(x)$ ($j \in \{1, \ldots, r\}$).

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

$$U_N(x)d_N < 0 \quad \text{and} \quad d_i \geq 0, \quad \forall i \in I(x) \cap N.$$

The first efficiency condition of Theorem 2.1 then may be reduced to:

Corollary 3.1 *A necessary condition for a nondegenerate point $x^* \in S$ to be locally weakly efficient solution for (MOP) is that*

$$\nexists d_N \in \mathbb{R}^{n-m} \ \text{ a multiobjective reduced descent direction of } F \text{ at } \ x^*.$$

In the next subsection, we will show how multiobjective reduced descent directions may be computed.

3.2 RJM Direction

In this subsection, we propose reduced descent directions of the Wolfe's type for the (MOP). Consider a positive one-dimensional 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, $\phi_0 = \mathbb{1}_{\mathbb{R}^*}$ 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 $\lfloor a \rfloor_+ = \max(0, a)$ be the positive part of the scalar a, and, $\lfloor a \rfloor_- = \max(0, -a)$ be its negative part.

Then, we introduce, for each $x \in S$, the following scalar subproblem:

$$(\mathcal{P}_x) \qquad \text{Min}_{\lambda \in \Lambda} \ f(\lambda, x) := \frac{1}{2} \sum_{i \in N} \left(\left\lfloor \sum_{j=1}^{r} \lambda_j u_i^j(x) \right\rfloor_-^2 + \phi(x_i) \left\lfloor \sum_{j=1}^{r} \lambda_j u_i^j(x) \right\rfloor_+^2 \right),$$

where $\Lambda = \left\{ (\lambda_1, \ldots, \lambda_r) \in \mathbb{R}_+^r : \sum_{j=1}^{r} \lambda_j = 1 \right\}$, λ_j is the jth component of λ, x_i is of course the ith component of the nonbasic vector x_N, and $u_i^j(x)$ is the ith component of the jth reduced gradient $u_N^j(x)$ of the objective f_j at x.

For $\lambda = (\lambda_1, \ldots, \lambda_r)$, we denote by $\delta_N(\lambda, x) = \Big[\delta_i(\lambda, x)\Big]_{i \in N}$ the vector mapping whose ith component, for each $i \in N$, is defined by

$$\delta_i(\lambda, x) = \left\lfloor \sum_{j=1}^{r} \lambda_j u_i^j(x) \right\rfloor_- - \phi(x_i) \left\lfloor \sum_{j=1}^{r} \lambda_j u_i^j(x) \right\rfloor_+.$$

Then, the functional $f(\cdot, x)$ satisfies the following good properties:

Lemma 3.1 *The functional $\lambda \mapsto f(\lambda, x)$ is positive, $\mathcal{C}^1$ and convex. Moreover, its gradient is given by*

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

Proof Clearly, $f(\lambda, x) \geq 0$ by its definition and the positivity assumption on ϕ. Also, the convex functionals $\lfloor\cdot\rfloor_-^2$ and $\lfloor\cdot\rfloor_+^2$ being composed with a linear form, and $\phi \geq 0$, the convexity of $f(\cdot, x)$ then follows by convexity of conic combinations of convex functions. The $\mathcal{C}^1$ property of $f(\cdot, x)$ follows by the same arguments. Now for computing $\nabla_\lambda f(\lambda, x)$, recall that the functionals $\lfloor\cdot\rfloor_-^2$ and $\lfloor\cdot\rfloor_+^2$ are derivable on $\mathbb{R}$ with derivatives $\left(\lfloor\cdot\rfloor_-^2\right)' = -2\lfloor\cdot\rfloor_-$ and $\left(\lfloor\cdot\rfloor_+^2\right)' = 2\lfloor\cdot\rfloor_+$. So, for each $k \in \{1, \ldots, r\}$,

$$\begin{aligned}\frac{\partial f}{\partial \lambda_k}(\lambda, x) &= -\sum_{i\in N}\left(\left\lfloor \sum_{j=1}^{r} \lambda_j u_i^j(x)\right\rfloor_- - \phi(x_i)\left\lfloor \sum_{j=1}^{r} \lambda_j u_i^j(x)\right\rfloor_+\right)u_i^k(x)\\ &= -\sum_{i\in N}\delta_i(\lambda, x)u_i^k(x)\\ &= -\delta_N(\lambda, x)\cdot u_N^k(x).\end{aligned}$$

Thus, $\nabla_\lambda f(\lambda, x) = -\left[\delta_N(\lambda, x)\cdot u_N^j(x)\right]_{j=1\cdots r} = -U_N(x)\delta_N(\lambda, x).$ □

Hence, $(\mathcal{P}_x)$ is a differentiable convex program having at least one optimal solution on the convex compact set Λ, say $\lambda(x) \in \Lambda$. Furthermore, its solutions may be characterized by the minimum variational principle as follows:

Lemma 3.2 *$\lambda(x) \in \Lambda$ is an optimal solution of $(\mathcal{P}_x)$, iff*

$$\delta_N\big(\lambda(x), x\big)\cdot\left[U_N(x)^{\mathrm{T}}\lambda\right] \leq -2f\big(\lambda(x), x\big) \qquad (\forall\lambda \in \Lambda),$$

or equivalently,

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

Proof Since $(\mathcal{P}_x)$ is a differentiable convex program, then by the minimum principle, $\lambda(x) \in \mathrm{argmin}(\mathcal{P}_x)$, if and only if

$$\nabla_\lambda f\big(\lambda(x), x\big)\cdot\big(\lambda - \lambda(x)\big) \geq 0 \qquad (\forall\lambda \in \Lambda).$$

Using the expression of $\nabla_\lambda f\big(\lambda(x), x\big)$ stated in Lemma 3.1, this inequality becomes equivalent to

$$\left[U_N(x)\delta_N(\lambda(x), x)\right]\cdot\lambda \leq \left[U_N(x)\delta_N(\lambda(x), x)\right]\cdot\lambda(x),$$

i.e.,

$$\delta_N\big(\lambda(x), x\big)\cdot\left[U_N(x)^{\mathrm{T}}\lambda\right] \leq \delta_N\big(\lambda(x), x\big)\cdot\left[U_N(x)^{\mathrm{T}}\lambda(x)\right].$$

Then, from the properties $a = \lfloor a \rfloor_+ - \lfloor a \rfloor_-$ and $\lfloor a \rfloor_+ \times \lfloor a \rfloor_- = 0$, it follows, by the very definition of δ_N, that

$$\begin{aligned}\delta_N\big(\lambda(x), x\big) \cdot \Big[U_N(x)^{\mathrm{T}}\lambda(x)\Big] &= \sum_{i \in N}\left(\delta_i\big(\lambda(x), x\big)\sum_{j=1}^{r}\lambda_j(x)u_i^j(x)\right)\\ &= \sum_{i \in N}\delta_i\big(\lambda(x), x\big)\Bigg(\Bigg\lfloor\sum_{j=1}^{r}\lambda_j(x)u_i^j(x)\Bigg\rfloor_+ - \Bigg\lfloor\sum_{j=1}^{r}\lambda_j(x)u_i^j(x)\Bigg\rfloor_-\Bigg)\\ &= \sum_{i \in N}\Bigg(-\Bigg\lfloor\sum_{j=1}^{r}\lambda_j(x)u_i^j(x)\Bigg\rfloor_-^2 - \phi(x_i)\Bigg\lfloor\sum_{j=1}^{r}\lambda_j(x)u_i^j(x)\Bigg\rfloor_+^2\Bigg)\\ &= -2\, f\big(\lambda(x), x\big),\end{aligned}$$

which proves the first inequality of the lemma. The second inequality follows necessarily, for each $j \in \{1 \ldots, r\}$, by taking successively $\lambda = e^j$ $(\in \Lambda)$, where e^j is the jth vector of the canonical basis of $\mathbb{R}^r$. Their sufficiency follows easily using the fact that $\lambda_j \geq 0$ $(j = 1 \cdot\cdot r)$ and $\sum\limits_{j=1}^{r} \lambda_j = 1$. □

Let $d_N(x) := \delta_N(\lambda(x), x)$, where $\lambda(x) = (\lambda_1(x), \ldots, \lambda_r(x)) \in \operatorname{argmin}(\mathcal{P}_x)$ with $x \in S$. Then,

$$d_N(x) = \left(\Bigg\lfloor\sum_{j=1}^{r}\lambda_j(x)u_i^j(x)\Bigg\rfloor_- - \phi(x_i)\Bigg\lfloor\sum_{j=1}^{r}\lambda_j(x)u_i^j(x)\Bigg\rfloor_+\right)_{i \in N}, \tag{5}$$

so that its components are explicitly given by: $\forall i \in N$,

$$d_i(x) = \begin{cases} -\phi(x_i)\sum\limits_{j=1}^{r}\lambda_j(x)u_i^j(x), & \text{if} \quad \sum\limits_{j=1}^{r}\lambda_j(x)u_i^j(x) > 0,\\ -\sum\limits_{j=1}^{r}\lambda_j(x)u_i^j(x), & \text{else.}\end{cases} \tag{6}$$

The vector $d_N(x)$ possesses multiobjective descent properties:

Proposition 3.1 (i) $d_N(x) = 0 \Leftrightarrow f(\lambda(x), x) = 0$.

(ii) *$d_N(x) \neq 0 \Rightarrow d_N(x)$ is a multiobjective reduced descent direction of F at x. If furthermore $x_B > 0$ (i.e., x is nondegenerate) with $B = \{1, \ldots, n\}\backslash N$, then it is not Pareto KKT-stationary for (MOP).*

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

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

Proof (i) To show the direct implication, observe that $\phi \geq 0$, and therefore $d_N(x) = \delta_N(\lambda(x), x) = 0 \Rightarrow \Big\lfloor\sum_{j=1}^{r}\lambda_j(x)u_i^j(x)\Big\rfloor_- = \Big\lfloor\phi(x_i)\sum_{j=1}^{r}\lambda_j(x)u_i^j(x)\Big\rfloor_+$. If we suppose that the last two terms are not equal to zero, we would have both

$\sum_{j=1}^{r} \lambda_j(x)u_i^j(x) < 0$ and $\phi(x_i)\sum_{j=1}^{r} \lambda_j(x)u_i^j(x) > 0$, which contradicts $\phi \geq 0$. Thus,

$$\left\lfloor \sum_{j=1}^{r} \lambda_j(x)u_i^j(x) \right\rfloor_- = \left\lfloor \phi(x_i)\sum_{j=1}^{r} \lambda_j(x)u_i^j(x) \right\rfloor_+ = 0,$$

and this shows that $f(\lambda(x), x) = 0$. The reverse implication follows similarly using once again the positivity of the mappings $\lfloor\cdot\rfloor_+$, $\lfloor\cdot\rfloor_-$ and ϕ, and the property: $\phi(t) = 0$ iff $t = 0$.

(ii) If $d_N(x) \neq 0$, then since $f(\lambda(x), x) \geq 0$ (see Lemma 3.1), it follows, by Proposition 3.1(i), that $f(\lambda(x), x) > 0$. Hence, using Lemma 3.2, we obtain that

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

or equivalently, $U_N(x)d_N(x) < 0$. On the other hand, $\phi(x_i) = 0$ if $i \in I(x)$. Hence, for $i \in I(x) \cap N$,

$$d_i(x) = \delta_i\big(\lambda(x), x\big) = \left\lfloor \sum_{j=1}^{r} \lambda_j(x)u_i^j(x) \right\rfloor_- \geq 0.$$

Thus, $d_N(x)$ is a multiobjective reduced descent direction of F at x. When $x_B > 0$, $I(x) = I(x) \cap N$, and Corollary 3.1 then becomes equivalent to Theorem 2.1, which affirms the conclusion of this assertion.

(iii) Suppose that $d_N(x) = 0$. Put $u_N^* := \sum_{j=1}^{r} \lambda_j(x)u_N^j(x) = U_N(x)^{\mathrm{T}}\lambda(x)$. Then by (6), it follows that $u_N^* \geq 0$, and if $u_i^* > 0$, then $x_i = 0$, because $\phi(x_i) = 0$. In other words,

$$u_N^* \geq 0, \quad \text{and} \quad u_N^* \cdot x_N = 0.$$

So it is easily seen that for $\lambda^* := \lambda(x) \in \mathbb{R}_+^r\backslash\{0\}$, $u^* := (0, u_N^*) \in \mathbb{R}_+^n$ and $v^* := -(A_B^{-1})^{\mathrm{T}} \sum_{j=1}^{r} \lambda_j(x)\dfrac{\partial f_j}{\partial x_B}(x) = -(A_B^{-1})^{\mathrm{T}} JF_B(x)^{\mathrm{T}}\lambda^* \in \mathbb{R}^m$, it holds that $u^* = JF(x)^{\mathrm{T}}\lambda^* + A^{\mathrm{T}}v^*$ and $u^* \cdot x = 0$ (since $u_B^* = 0$), which express the Pareto KKT-stationarity for (MOP).

(iv) Since $f(\lambda(x), x)$ is the optimal value of $(\mathcal{P}_x)$, its continuity is directly coming from the compactness of the constraint set Λ of $(\mathcal{P}_x)$ and the continuity of its objective function $(\lambda, x) \mapsto f(\lambda, x)$.

□

3.3 Special RJM Directions

There are several manners to construct reduced descent directions depending on the choice of the functionals ϕ defined in Sect. 3.2. Taking, for example, $\phi = \phi_0$, the multiobjective reduced direction given by (6) has the following form: $\forall i \in N$,

$$d_i(x) = \begin{cases} 0, & \text{if } x_i = 0 \text{ and } \sum\limits_{j=1}^{r} \lambda_j(x) u_i^j(x) > 0, \\ -\sum\limits_{j=1}^{r} \lambda_j(x) u_i^j(x), & \text{else.} \end{cases}$$

In the scalar case ($r = 1$), this scheme coincides with the discontinuous scheme of Wolfe: $\forall i \in N, d_i(x) = 0$ if $x_i = 0$ and $u_i(x) > 0, d_i(x) = -u_i(x)$ otherwise, where $u_i(x)$ stands for the ith component of the reduced gradient of the single objective to be minimized. This scheme may be written under a compact form: $d_i(x) = \lfloor u_i(x) \rfloor_- - \mathbb{1}_{\mathbb{R}^*}(x_i) \lfloor u_i(x) \rfloor_+$.

Another important case is to take $\phi = \phi_p$ with $p = 1$. Since it is assumed that x is feasible (in particular, $x \geq 0$), this leads, according to (6), to the following multiobjective scheme: $\forall i \in N$,

$$d_i(x) = \begin{cases} -x_i \sum\limits_{j=1}^{r} \lambda_j(x) u_i^j(x), & \text{if } \sum\limits_{j=1}^{r} \lambda_j(x) u_i^j(x) > 0, \\ -\sum\limits_{j=1}^{r} \lambda_j(x) u_i^j(x), & \text{else,} \end{cases} \tag{7}$$

which also reduces in the scalar case to the Wolfe's continuous scheme: $\forall i \in N$, $d_i(x) = -x_i\, u_i(x)$ if $u_i(x) > 0$, $d_i(x) = -u_i(x)$ otherwise. Its compact form: $d_i(x) = \lfloor u_i(x) \rfloor_- - x_i \lfloor u_i(x) \rfloor_+$ clearly shows its continuity. The reader can consult the books [14,15] or the recent paper [16] for more details on RGM.

4 RJM Algorithm

4.1 Multiobjective Stepsize

To compute a descent steplength along a descent direction d satisfying (1) at the current point x, i.e., $JF(x)d < 0$, we use an extended Armijo rule [24] for vector mappings by Fliege and Svaiter [9]:

Proposition 4.1 *Let $x, d \in \mathbb{R}^n$ and $\beta \in \,]0, 1[$. If $JF(x)d < 0$, then*

$$\exists\, t_a > 0, \quad \forall t \in \,]0, t_a\,], \qquad F(x + td) < F(x) + \beta\, t\, JF(x)d. \tag{8}$$

Any fixed scalar $t = t_0 > 0$ may be a starting guess, and while the vector inequality (8) is not satisfied, the steplength t is reduced to $t/2$. However, to ensure a feasible descent direction for the next iterate $x + td$, the choice of the initial guess is very important for the computation. As seen before in the reduced gradient strategy, $Ad = 0$ by construction, since $d_B = -A_B^{-1} A_N d_N$. So only the positivity constraint remains to be checked for the next iterate $x + td$. When the direction $d \geq 0$, the positivity is preserved for any $t \in \mathbb{R}_+$, a situation that may occur only if the feasible set is unbounded. In this case, the Armijo rule may be started with $t_0 = 1$. Otherwise ($d \ngeq 0$), the following upper bound of stepsizes:

$$t_f = \min\left\{\frac{-x_i}{d_i} \;:\; d_i < 0,\; i = 1 \cdots n\right\} \tag{9}$$

may be a suitable initial guess. Indeed, we have the following nice properties:

Lemma 4.1 *Let $x \in S$ and $d \ngeq 0$ be such that $d_B = -A_B^{-1} A_N d_N$ with d_N given by (6). Denote $x(t) = x + td$ with $t \in \mathbb{R}_+$. Then,*

(i) $t \leq t_f \;\Leftrightarrow\; x(t) \geq 0$.
If furthermore $x_B > 0$ (i.e., x is nondegenerate), then
(ii) $t_f > 0$*;*
(iii) $t < t_f \;\Rightarrow\; x_B(t) > 0$ *(i.e., $x(t)$ is nondegenerate); $t = t_f \;\Rightarrow\; \exists i \in \{1, \ldots, n\}$: $x_i(t) = 0$.*

Proof (i) It is clear that t_f exists (since $d \ngeq 0$) and $t_f \geq 0$. Let $i \in \{1, \ldots, n\}$. Then $x_i(t) \geq 0$ if $d_i \geq 0$; otherwise, $d_i < 0$ implies that $-x_i/d_i \geq t_f \geq t$, so that $0 \leq x_i + td_i = x_i(t)$. Conversely, let $i \in \{1, \ldots, n\}$ be such that $d_i < 0$. Then $x_i(t) \geq 0$ implies that $t \leq -x_i/d_i$, and this shows that $t \leq t_f$.

(ii) According to (9), let $t_f = -x_i/d_i$ with $d_i < 0$ for some $i \in \{1, \ldots, n\}$. If $i \in B$, then $x_i > 0$ by the assumption, which shows that $t_f > 0$. If $i \in N$, then by virtue of (6), $d_i = -\phi(x_i) \sum_{j=1}^{r} \lambda_j(x) u_i^j(x) < 0$ implies that $\phi(x_i) > 0$, which in turn implies that $x_i > 0$, showing still again that $t_f > 0$.

(iii) Let $i \in B$. Then, by the nondegeneracy assumption of x, $x_i(t) > 0$ if $d_i \geq 0$; otherwise $t_f \leq -x_i/d_i$, so that $0 < x_i + td_i = x_i(t)$ for any $t < t_f$. If $t = t_f = -x_i/d_i$ for some $i \in \{1, \ldots, n\}$, then $x_i(t) = 0$, and this completes the proof of the lemma. □

4.2 The Algorithm Outline

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

Remark 4.1 1. The nondegeneracy hypothesis on S theoretically leaves Step 1 achievable. In the worst case, we need to find a new base B at each iteration of the algorithm, but in practice it is not often needed (see Sect. 5).

2. In Step 4, every optimal solution $\lambda(x)$ of $(\mathcal{P}_x)$, which lies in $\text{int}\mathbb{R}^r_+$, will indicate, in the case of a convex (MOP), that $x \in E_p$ [see Remark 2.2 and proof of Proposition 3.1(iii)].
3. In Step 5, by virtue of (4), we simply use $U_N d_N$ in the Armijo condition (8) instead of $JF(x)d$ (avoiding extra computations). On the other hand, observe that if x was degenerate with some $i \in B$ such that $d_i < 0$, then we would have $t_f = 0$, and according to Lemma 4.1(i), this step would not be achieved, and the algorithm would not be well defined.
4. In Step 7, according to Lemma 4.1(iii), degeneracy happens for the updated point $x(t) = x + td$, when t coincides with t_f, that attains its minimum at a basic index. So the nondegenerate basis selection (Step 1) is needed in this case; otherwise, nondegeneracy of x is conserved for the next iterate $x(t)$.

RJM 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 $\phi$.
Step 1: Nondegenerate basis selection.
– Identify a basis $B$ such that $x_B > 0$. Set $N = \{1, \ldots, n\} \backslash B$.
Step 2: Reduced Jacobian.
– Compute the reduced Jacobian matrix $U_N$ at $x$.
Step 3: Reduced direction.
– Solve $(\mathcal{P}_x)$ to determine $d_N$ according to (6).
Step 4: Stopping criterion.
– If $\min(\mathcal{P}_x) = 0$, then stop: $x$ is Pareto KKT-stationary for (MOP).
Step 5: Armijo's line search.
– Define $d_B = -A_B^{-1} A_N d_N$ and set $d = (d_B, d_N)$.
– Compute $t_f$ according to (9).
– Determine a steplength $t \in ]0, t_f]$ as $t = \max_{p \in \mathbb{N}} \left\{ \frac{1}{2^p} : \quad F\left(x + \frac{1}{2^p} d\right) < F(x) + \beta \frac{1}{2^p} U_N d_N \right\}$.
Step 6: Updated point.
– Set $x := x + td$.
Step 7: Degeneracy test.
– If $t = t_f = -x_i/d_i$ for some $i \in B$, go to Step 1; otherwise go to Step 2.
```

4.3 Global Convergence

If the RJM algorithm is finite, then according to Proposition 3.1, it terminates at a Pareto KKT-stationary point for (MOP). Thus, we will assume that the algorithm is infinite. The following theorem gives a global convergence result. Its proof closely follows the convergence lines of methods that use the Armijo's condition (see, e.g., [16] for the linearly constrained single-objective case, or [9] for the unconstrained multiobjective case).

Theorem 4.1 *Assume that the nondegeneracy assumption on S holds and ϕ is chosen to be continuous. Let $\left(x^k\right)_{k \in \mathbb{N}}$ be the sequence produced by the RJM algorithm. Then,*

(i) $\left(x^k\right)_k \subset S$, *and,* $F(x^{k+1}) < F(x^k)$ *for all* $k \in \mathbb{N}$;

(ii) *any accumulation point* x^* *of* $\left(x^k\right)_k$ *is a Pareto KKT-stationary point for (MOP), and,* $\lim_{k \to +\infty} F(x^k) = F(x^*)$.

Proof (i) The first part of this assertion is obvious from Lemma 4.1(i) and the fact that $Ad = 0$ (since $d_B = -A_B^{-1} A_N d_N$). Now by the Armijo condition, one has

$$\forall k \in \mathbb{N}, \qquad F(x^{k+1}) - F(x^k) < \beta\, t_k\, U_{N_k}(x^k) d_{N_k}(x^k) < 0. \tag{10}$$

The last inequality is due to Proposition 3.1(i)–(ii) since it is assumed that the algorithm is infinite, i.e., $f(\lambda(x^k), x^k) = \min(\mathcal{P}_{x^k}) > 0$ $(\forall k)$. Thus, we can see that the vector sequence $\left(F(x^k)\right)_{k \in \mathbb{N}}$ is decreasing.

(ii) By hypothesis, $\exists \left(x^k\right)_{k \in \mathcal{K}_1} \longrightarrow x^*$ with $\mathcal{K}_1 \subseteq \mathbb{N}$ an infinite subset. By closedness of the feasible set, x^* remains feasible as $\left(x^k\right)_k$. It suffices to prove, according to Proposition 3.1(i)–(iii), that $f(\lambda(x^*), x^*) = \min(\mathcal{P}_{x^*}) = 0$. The $\lim\limits_{k\to+\infty} F(x^k) = F(x^*)$ is a direct consequence of Theorem 4.1(i) and the continuity of F. Passing onto the limit in (10) as $k \nearrow +\infty$, we deduce that

$$\lim_{k\to+\infty} t_k\, U_{N_k}(x^k) d_{N_k}(x^k) = 0.$$

Since the index set B_k (resp. $N_k = \{1, \dots, n\}\backslash B_k$) of the basic (resp. nonbasic) components of x^k is in the finite set $\{1, \dots, n\}$, there exists an infinite subset $\mathcal{K}_2 \subseteq \mathcal{K}_1$ such that $B_k = B$ (hence $N_k = N$) $\forall k \in \mathcal{K}_2$. In particular, one obtain that

$$\lim_{\substack{k\in\mathcal{K}_2\\ k\to+\infty}} t_k\, U_N(x^k) d_N(x^k) = 0. \tag{11}$$

But in sight of (5), $\big(d_N(x^k)\big)_{k\in\mathcal{K}_2}$ is a bounded sequence because $\big(\lambda(x^k)\big)_k \subset \Lambda$ is bounded, and $\left(u_i^j(x^k)\right)_k$ and $\big(\phi(x^k)\big)_k$ are, by continuity, convergent. So by considering a suitable subsequence $\big(d_N(x^k)\big)_{k\in\mathcal{K}_3} \longrightarrow d_N^*$ with $\mathcal{K}_3 \subseteq \mathcal{K}_2$, it follows that the negative sequence $\big(U_N(x^k)d_N(x^k)\big)_{k\in\mathcal{K}_3} \longrightarrow U_N(x^*)d_N^* \leq 0$ and that $\big(d_B(x^k)\big)_{k\in\mathcal{K}_3} = \left(-A_B^{-1}A_N d_N(x^k)\right)_{k\in\mathcal{K}_3} \longrightarrow -A_B^{-1}A_N d_N^* := d_B^*$. We shall prove that there exists some $j_0 \in \{1, \dots, r\}$ such that $\left(U_N(x^*)d_N^*\right)_{j_0} = 0$. Indeed, suppose by contradiction way that $U_N(x^*)d_N^* < 0$. Then, according to (11), $t_k \longrightarrow 0$. Hence, for any fixed $p \in \mathbb{N}$ and all $k \in \mathcal{K}_3$ large enough, $t_k < 1/2^p$. This means that the Armijo's condition would not be satisfied at $t = 1/2^p$ for all $k \in \mathcal{K}_3$ sufficiently large:

$$F\left(x^k + \frac{1}{2^p} d(x^k)\right) \not\leq F(x^k) + \beta \frac{1}{2^p} U_N(x^k) d_N^k.$$

By letting $k \nearrow +\infty$, we would obtain

$$f_{j_p}\left(x^* + \frac{1}{2^p} d^*\right) \geq f_{j_p}(x^*) + \beta \frac{1}{2^p} \left(U_N(x^*)d_N^*\right)_{j_p}, \tag{12}$$

for some $j_p \in \{1, \dots, r\}$, where $d^* = (d_B^*, d_N^*)$. Hence, by (4), it follows that $JF(x^*)d^* = U_N(x^*)d_N^*$. Since it was assumed that $U_N(x^*)d_N^* < 0$ and $p \in \mathbb{N}$ was arbitrary, the previous inequality (12) would lead to a contradiction with Proposition 4.1. Therefore $\exists j_0 \in \{1, \dots, r\}$:

$$0 = \left(U_N(x^*)d_N^*\right)_{j_0} = \lim_{\substack{k\in\mathcal{K}_3\\ k\to+\infty}} \left(U_N(x^k)d_N(x^k)\right)_{j_0}.$$

Now, according to Lemma 3.2, one has

$$\left(U_N(x^k)d_N(x^k)\right)_{j_0} = d_N(x^k) \cdot u_N^{j_0}(x^k) \leq -2\, f\left(\lambda(x^k), x^k\right) < 0.$$

So letting $k \nearrow +\infty$, by Proposition 3.1(iv), we get $0 = f(\lambda(x^*), x^*)$, and according to Proposition 3.1(i)-(iii), we conclude that x^* is well a Pareto KKT-stationary point for (MOP). □

Remark 4.2 An accumulation point of the generated sequence $\left(x^k\right)_k$ exists, for instance, if the feasible level subset $\{x \geq 0 : Ax = b,\ F(x) \leq F(x^0)\}$ is bounded, since it contains $\left(x^k\right)_k$. Hence, if such a point exists, then by replacing the stopping criterion "$\min(\mathcal{P}_x) = 0$" by "$\min(\mathcal{P}_x) \leq \varepsilon$" in the RJM algorithm, with any prescribed tolerance $\varepsilon > 0$, the algorithm will stop, according to Theorem 4.1, after a finite number of iterations.

5 Numerical RJM Experiments

This section exposes some numerical experiments on our RJM described in Sect. 4.2. The code is written in MATLAB 7.10 and run on a personal computer equipped with the Windows 7, 1.33GHz Intel Core 2 Duo CPU and 2GB memory. We tested our algorithm on various test problems from the literature, in addition to two problems that we suggest here. We restricted ourselves to only biobjective problems in order to graphically investigate local and/or global Pareto fronts, as we also think that two objectives are sufficient to reflect essential aspects of the algorithm. This, however, does not exclude large-scale dimensionally problems. All those examples are given here with their original feasible sets.

To highlight the advantages of RJM, and since one of the main objectives of this paper is to avoid scalarization, we also compared RJM with RGM applied on scalarized problems, denoted here by W-RGM, which is not other than RJM applied to single-objective programs from the weighting method of scalarization with scalarized function, $F_\lambda : \mathbb{R}^n \to \mathbb{R}$ given by

$$F_\lambda(x) = \lambda f_1(x) + (1-\lambda) f_2(x), \qquad 0 \leq \lambda \leq 1.$$

Test 1 We begin by a 1-dimensional biobjective problem:

$$\begin{array}{ll} \text{Minimize} & f_1(x) = \cos(4x)\sin(x),\ \ f_2(x) = \cos(4x)\cos(x), \\ \text{subject to} & 0 \leq x \leq 7. \end{array}$$

This problem has Pareto optima with very diversified local solutions.

Test 2 We consider now a 2-dimensional biobjective problem:

$$\begin{array}{ll} \text{Minimize} & f_1(x_1,x_2) = (x_1-x_2)^3,\ \ f_2(x_1,x_2) = x_1+x_2, \\ \text{subject to} & x_1+x_2 \leq 1,\ \ x_1-x_2 \leq 0.4,\ \ x_i \geq 0,\ \ i=1,2. \end{array}$$

This problem is pseudoconvex and then has only global Pareto minima.

Test 3 This 2-dimensional biobjective problem is taken from [4]:

$$\begin{array}{ll}\text{Minimize} & f_1(x_1,x_2)=x_1,\ f_2(x_1,x_2)=\dfrac{1+x_2}{x_1},\\ \text{subject to} & -9x_1-x_2\le -6,\ -9x_1+x_2\le -1,\ 0.1\le x_1\le 1,\ 0\le x_2\le 5.\end{array}$$

This problem has a convex Pareto front and hence only global Pareto minima.

Test 4 This 2-dimensional biobjective problem is located in [10]:

$$\begin{array}{ll}\text{Minimize} & f_1(x_1,x_2)=-x_1^2+x_2,\ f_2(x_1,x_2)=0.5x_1+x_2+1,\\ \text{subject to} & \frac{1}{6}x_1+x_2\le 6.5,\ 0.5x_1+x_2\le 7.5,\ 5x_1+x_2\le 30,\\ & x_i\ge 0,\ i=1,2.\end{array}$$

This problem is not pseudoconvex (take $x=(1,2)$ and $x'=(0,0)$ in Definition 2.1), hence nonconvex, but has only global Pareto minima. In fact, its objective F is concave ($-F$ is convex).

The following three test examples, known as ZDT problems [25], have the same generic form and consist of m-dimensional two objectives that we tested for various values of m:

$$\begin{array}{ll}\text{Minimize} & f_1(x)=x_1,\ f_2(x)=g(x)h\left(x_1,g(x)\right),\\ \text{with} & g(x)=1+\dfrac{9}{m-1}\displaystyle\sum_{i=2}^{m}x_i,\\ \text{subject to} & 0\le x_i\le 1,\ i=1,\ldots m.\end{array}$$

These problems differ only in h:

Test 5 In the first ZDT problem, h is given by:

$$h\left(x_1,g(x)\right)=1-\sqrt{\frac{x_1}{g(x)}}.$$

The Pareto front of this problem is convex and formed with $g(x)=1$.

Test 6 In the second ZDT problem, h is given by:

$$h\left(x_1,g(x)\right)=1-\left(\frac{x_1}{g(x)}\right)^2.$$

This problem is the nonconvex counterpart to *Test 5*. Its nonconvex Pareto front is still formed with $g(x)=1$.

Test 7 In the third ZDT problem, h is given by:

$$h\left(x_1, g(x)\right) = 1 - \sqrt{\frac{x_1}{g(x)}} - \left(\frac{x_1}{g(x)}\right)\sin(10\pi x_1).$$

This problem is nonconvex and has several discrete convex local parts in its Pareto front, which is still formed with $g(x) = 1$. This is due to the introduction of the sine function in h that causes discontinuity in border regions including Pareto sections.

In all the figures presented below, we put the RJM algorithm on left and W-RGM on right. In these figures, the term "convergence path" designates the route that the sequence produced by the algorithm takes to reach a Pareto point.

Except for *Test 1*, all other problems were solved with a population of $Q = 200$ individuals (starting points). This test problem was solved with $Q = 400$ individuals because of its particularity to possess very diverse local Pareto minima dispersed in various zones of the feasible image set. For each test problem, the comparison between RJM and W-RGM was done using the same selected population. Each feasible polyhedron is set under the standard form: $Ax = b,\ x \geq 0$. After a basis B is selected, nonbasic coordinates x_N are randomly chosen between a positive lower bound and an upper bound of the feasible polyhedron. The vector basic variable is of course computed according to (3), so that $Ax = b$, and while $x_B \not> 0$, the initialization procedure is repeated. For W-RGM, each qth individual, $q = 1, \ldots, Q$, has a different weight λ randomly chosen in an equidistant partition of [0, 1], namely $\left[\frac{q-1}{Q}, \frac{q}{Q}\right]$, in order to get them all scattered in different areas of the interval [0, 1].

During the numerical experiments, we prescribed $\beta = 0.25$ as Armijo's constant, and we started the Armijo procedure with the initial guess $t_0 = t_f$ instead of $t_0 = 1$. We used the scheme (7), which is continuous and avoids jamming. On the other hand, the convex differentiable program $(\mathcal{P}_x)$ is in turn solved by a standard RGM using the Wolfe's continuous scheme.

During running, the stopping criterion was: $\min(\mathcal{P}_x) \leq 10^{-6}$.

In Table 1 of numerical results, the following main characteristics of our investigated problems are reported:

Pb: name of the test problem.
m: number of the original variables.
n: number of the variables with standard form.
Iter: average number of iterations per population.
Feval: average number of vector function evaluations per population.
Beval: average number of basis changes per population.
Fig: figure reference of solutions by RJM and W-RGM.

From Table 1, we can see that the proposed method gives a solution to the linearly constrained (MOP) in few iterations with a reasonable number of function evaluations. Remark that, in almost all cases, RJM did not need any basis change. Also, the RJM scheme works well for large-scale problems, and it is not very sensitive to increased dimension n, not only from the point of view of computational costs, but also in the quality of the approximated Pareto fronts, as seen in the figures. On the other hand,

Table 1 Data and numerical results for *Tests 1–7*

Pb	*m*	*n*	Iter	Feval	Beval	Fig	
Test 1	1	2	0.84	6.08	0.05	1	RJM
			3.37	25.24	0.24		W-RGM
Test 2	2	4	0.99	1.99	0.55	2	RJM
			2.02	3.02	1.29		W-RGM
Test 3	2	6	1.00	2.00	0.61	3	RJM
			2.29	5.24	1.60		W-RGM
Test 4	2	5	1.00	2.00	0.55	4	RJM
			2.03	3.03	1.35		W-RGM
Test 5	50	100	2.75	7.84	0.00	–	RJM
			33.91	38.64	31.88		W-RGM
	100	200	2.81	9.01	0.00	–	RJM
			68.46	78.10	71.24		W-RGM
	200	400	2.94	9.31	0.00	5	RJM
			136.76	156.56	128.30		W-RGM
Test 6	50	100	1.00	2.00	0.00	–	RJM
			17.24	18.19	15.97		W-RGM
	100	200	1.00	2.00	0.00	–	RJM
			38.10	39.15	36.84		W-RGM
	200	400	1.00	2.00	0.00	6	RJM
			65.49	66.5	64.14		W-RGM
Test 7	50	100	2.98	14.00	0.00	–	RJM
			456.11	857.88	41.31		W-RGM
	100	200	2.86	12.41	0.00	–	RJM
			734.12	1621.6	98.11		W-RGM
	200	400	3.21	14.23	0.00	7	RJM
			1132.13	1720.1	187.32		W-RGM

we can see that the method is able to identify Pareto fronts for convex cases, and for all the other nonconvex cases, it is also able to discover all Pareto sections, whether they are global or local, while the population size used is relatively modest. Note also the ability of the algorithm to handle some serious difficulties, such as discontinuity or nonuniform density (see Figs. 1 and 7).

Comparing the results of RJM and W-RGM in Table 1 shows that W-RGM spent in both number of iterations, function evaluations and basis changes, especially for the ZDT problems. Note also that W-RGM did not completely recover the Pareto front even in the convex case, as clearly seen in Fig. 3 with *Test 3*. Moreover, from all figures corresponding to the nonconvex cases, including the pseudoconvex problem *Test 2*, we see that W-RGM has found only the extreme efficient points, and no local solution has been explored.

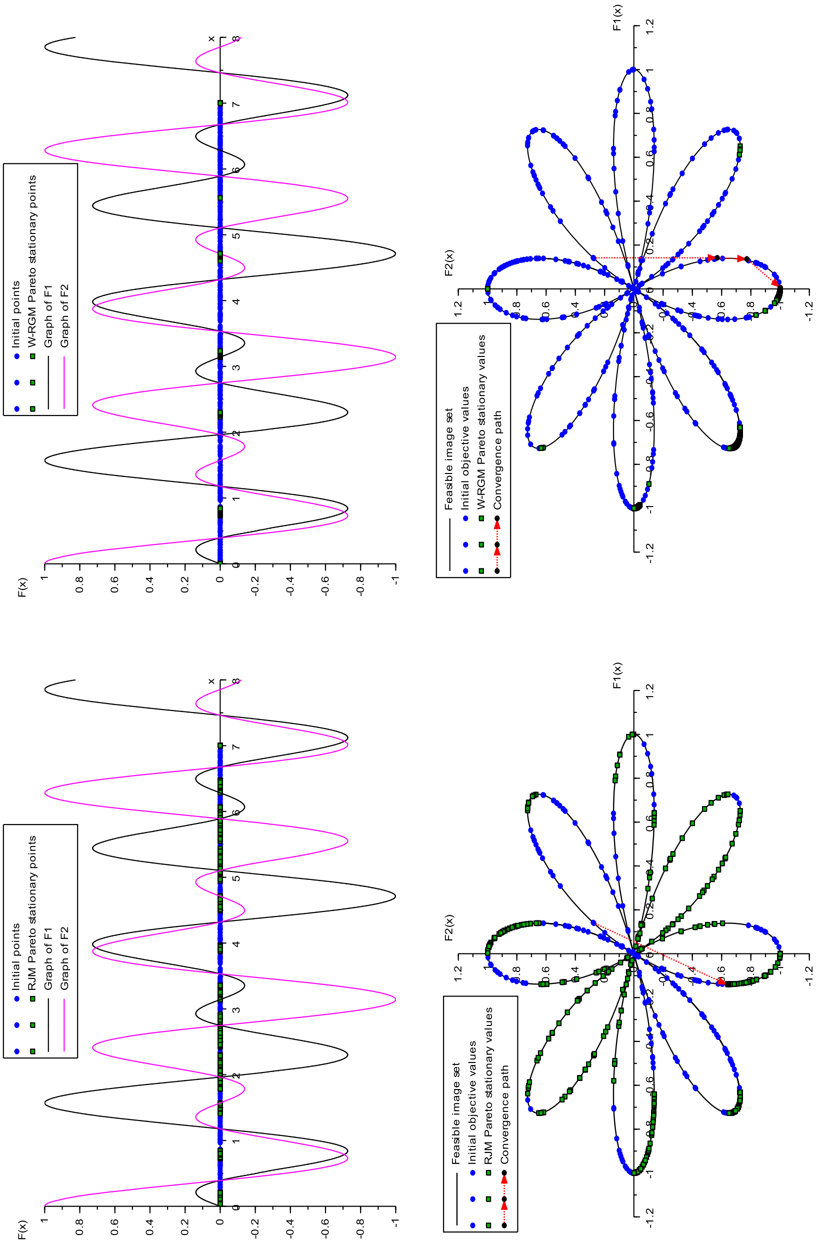

Fig. 1 Approximated Pareto set and Pareto front by RJM and W-RGM for *Test 1*

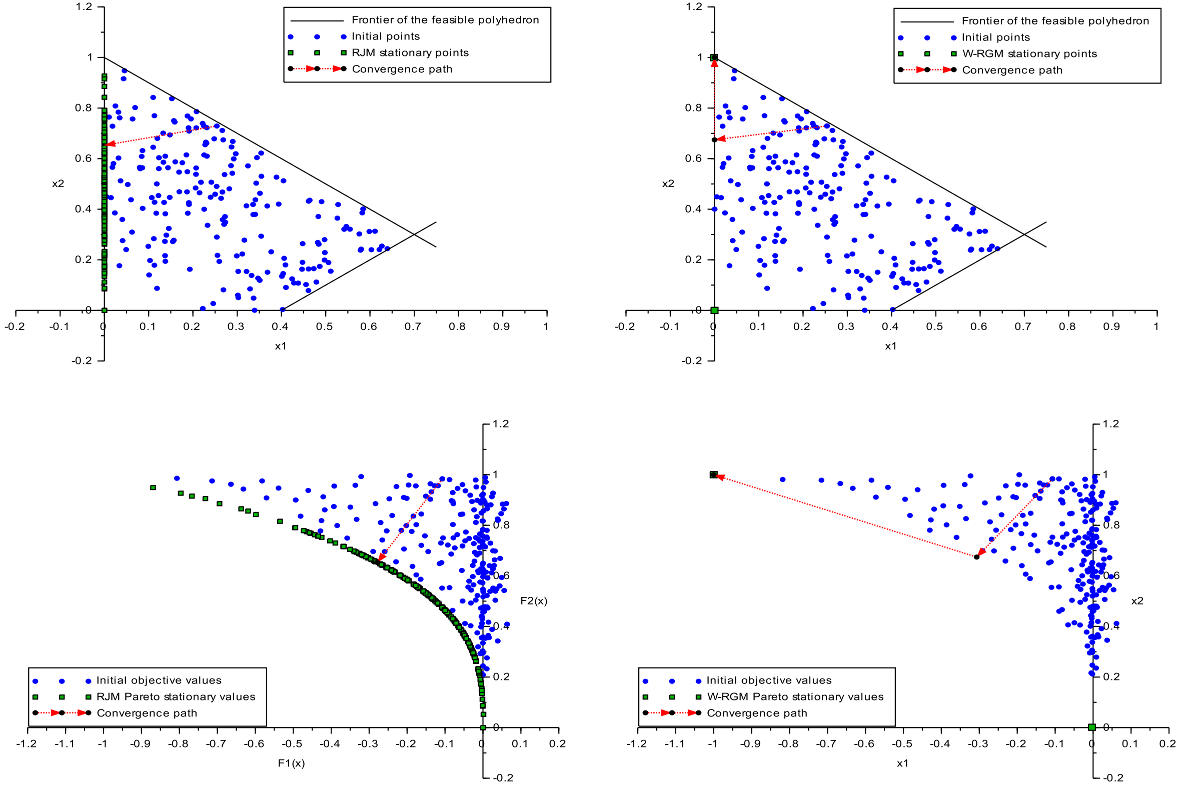

Fig. 2 Approximated Pareto set and Pareto front by RJM and W-RGM for *Test 2*

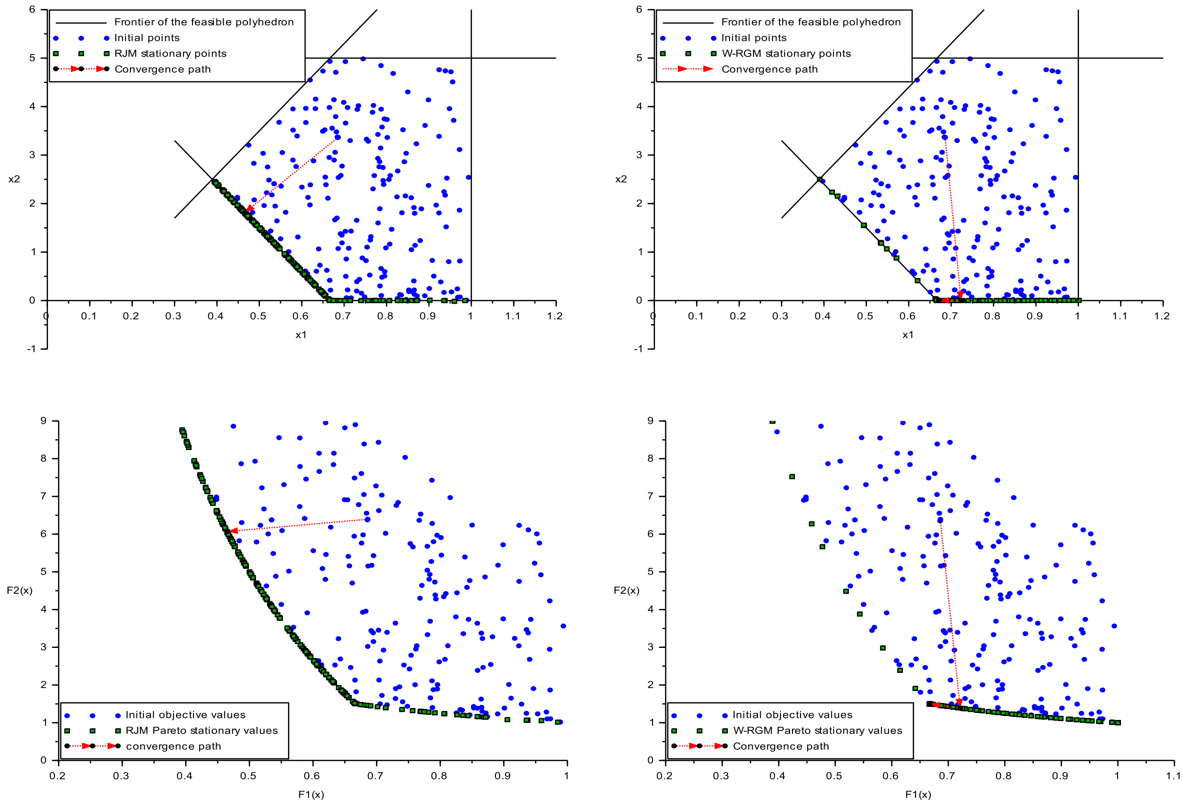

Fig. 3 Approximated Pareto set and Pareto front by RJM and W-RGM for *Test 3*

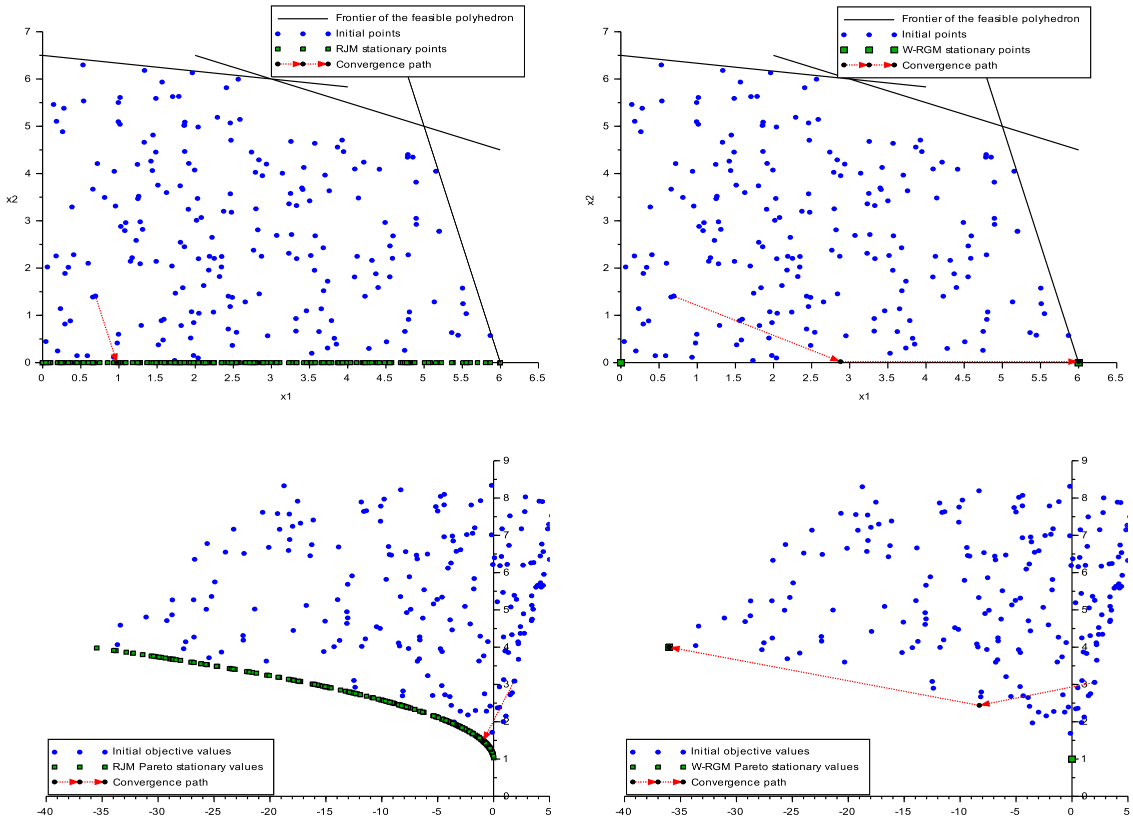

Fig. 4 Approximated Pareto set and Pareto front by RJM and W-RGM for *Test 4*

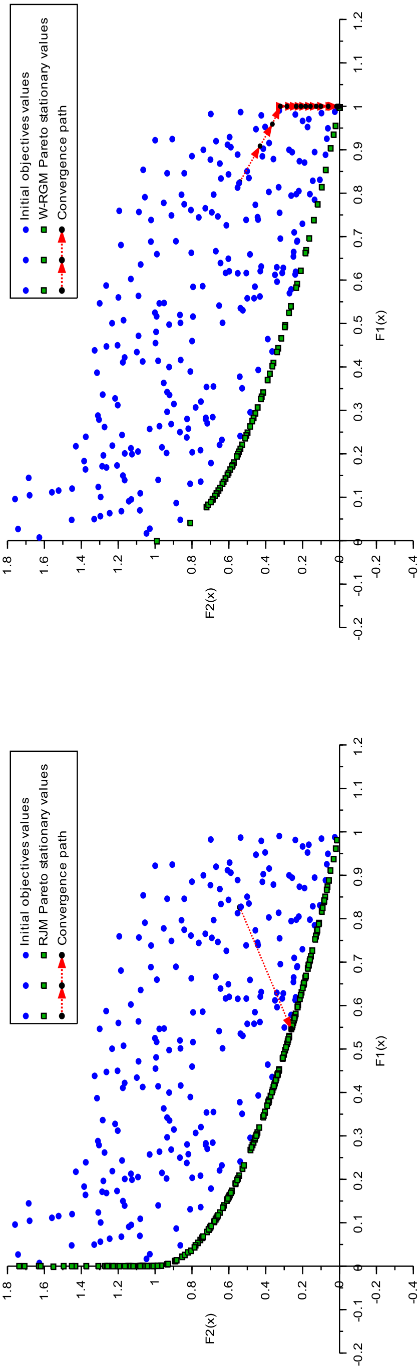

Fig. 5 Approximated Pareto front by RJM and W-RGM for *Test 5* with $m = 200$

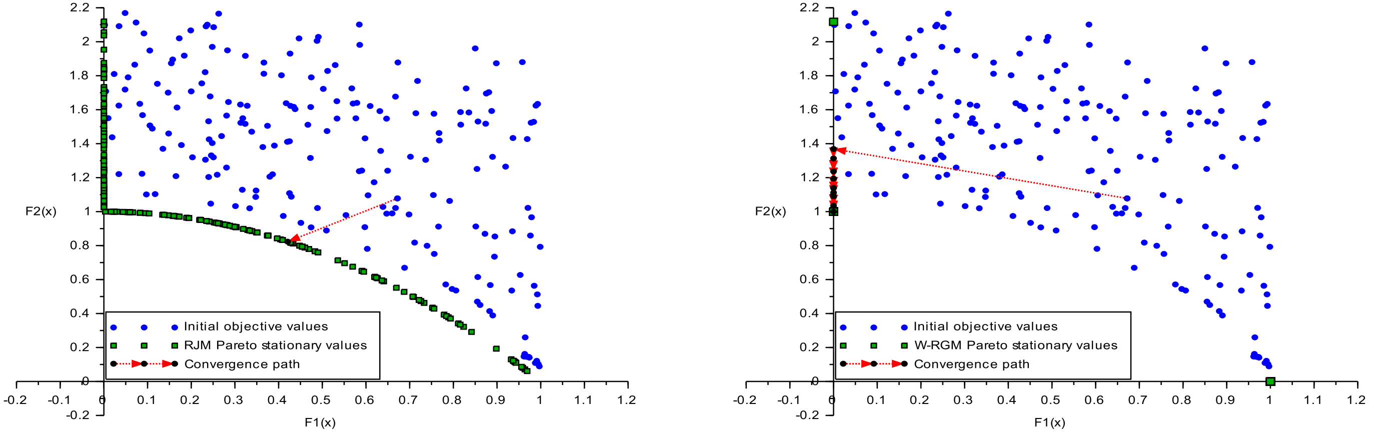

Fig. 6 Approximated Pareto front by RJM and W-RGM for *Test 6* with $m = 200$

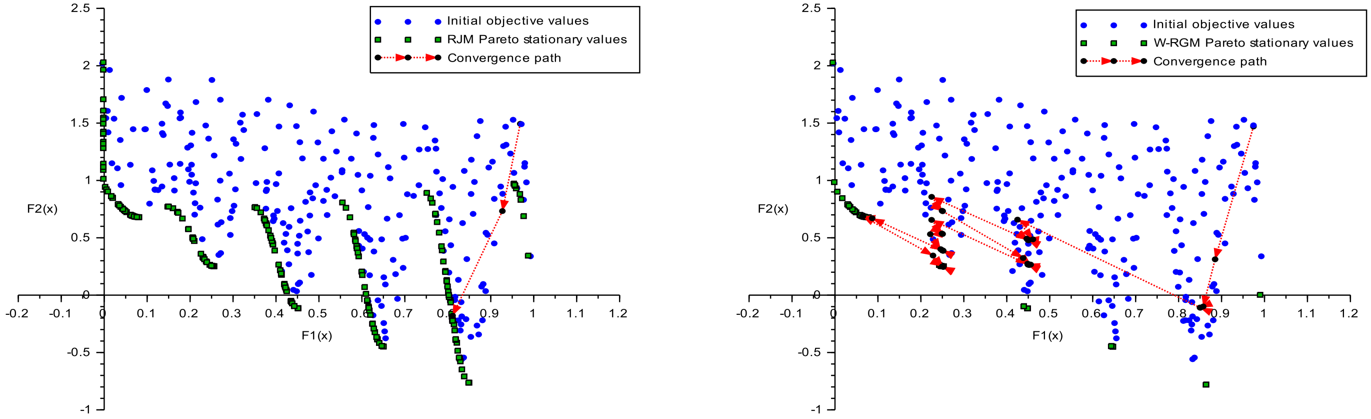

Fig. 7 Approximated Pareto front by RJM and W-RGM for *Test 7* with $m = 200$

6 Conclusions

In this paper, we presented a first extension of Wolfe's reduced gradient method from scalar to vector optimization to handle linear constraints. We called this method "Reduced Jacobian Method" (in brief, RJM) because the reduction principle is directly applied to the Jacobian matrix of the objective without scalarizing the vector problem. To this end, we proposed a descent direction that is common to all the objectives, by solving in the image space a simple scalar program having only one linear constraint and nice properties. It was also revealed that the optimal solutions of this program are not other than Lagrange multipliers associated with the objective of the (MOP), so that under some convexity assumptions, we can get more precisions about the kind of the efficient solutions obtained by RJM. We also provided a unified formula covering both the continuous and discontinuous schemes of Wolfe for single-objective programs. The (global) convergence of the resulting algorithm was proved under standard hypotheses, while just multiobjective Armijo economic stepsizes were used. Numerical experiments including large-scale problems were reported. A comparative study with the classical scalarization approach showed that this latter is far from being an efficient method for finding Pareto fronts, whereas the results obtained by RJM confirmed its good performances as expected regarding the well-known RGM ones. This may be explained by the fact that the method is conceived to find directly, and in an optimal way, a multiobjective feasible descent direction to the vector problem. Another important point is the fact that multiobjective problems have in principle a set of optimal values (Pareto front), so there are several possibilities to reach one of them, starting from any initial guess. It seems from the figures that RJM tries to find a short path toward an efficient value, and thus, the choice of well-dispersed populations helped our method to better explore the landscape of Pareto fronts.

Further research avenues may involve some applications, for example the important model of the so-called bilevel problem, which consists in minimizing a scalar function (e.g., cost or energy) over the Pareto front (see [26], for example). As this problem is known to be a difficult global optimization problem, it may be solved, for instance, by scalar metaheuristics. Taking into account that these evolutionary algorithms run on a set of feasible individuals and that their efficiency relies upon well selected populations, perhaps at a given iteration, we think that RJM could offer all tools to achieve such tasks.

Acknowledgements The authors are thankful for the valuable suggestions of the editor Dr. Franco Giannessi and the reviewers that helped improve the quality of the paper.

References

1. Collette, Y., Siarry, P.: Multiobjective Optimization. Principles and Case Studies. Springer, Berlin (2004)
2. Figueira, J., Greco, S., Ehrgott, M.: Multiple criteria decision analysis. In: State of the Art Surveys. Springer, New York (2005)
3. Ehrgott, M.: Multicriteria Optimization, 2nd edn. Springer, Berlin (2005)
4. Deb, K.: Multi-objective Optimization Using Evolutionary Algorithm. Wiley, Chichester (2001)

5. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer, Boston (1999)
6. Fliege, J., Graña Drummond, L.M., Svaiter, B.F.: Newton's method for multiobjective optimisation. SIAM J. Optim. **20**, 602–626 (2009)
7. Fliege, J.: An efficient interior-point method for convex multicriteria optimization problems. Math. Oper. Res. **31**, 825–845 (2006)
8. Eichfelder, G.: An adaptive scalarization method in multiobjective optimization. SIAM J. Optim. **19**, 1694–1718 (2009)
9. Fliege, J., Svaiter, B.F.: Steepest descent methods for multicriteria optimization. Math. Meth. Oper. Res. **51**, 479–494 (2000)
10. Graña Drummond, L.M., Iusem, A.N.: A projected gradient method for vector optimization problems. Comput. Optim. Appl. **28**, 5–29 (2004)
11. García-Palomares, U.M., Burguillo-Rial, J.C., González-Castaño, F.J.: Explicit gradient information in multiobjective optimization. Oper. Res. Lett. **36**, 722–725 (2008)
12. Bonnel, H., Iusem, A.N., Svaiter, B.F.: Proximal methods in vector optimization. SIAM J. Optim. **15**, 953–970 (2005)
13. 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)
14. Bazaraa, M.S., Sherali, H.D., Shetty, C.M.: Nonlinear Programming: Theory and Algorithms, 3rd edn. Wiley, New York (2006)
15. Luenberger, D.G., Ye, Y.: Linear and Nonlinear Programming, 3rd edn. Springer, New York (2008)
16. El Maghri, M.: A free reduced gradient scheme. Asian J. Math. Comput. Res. **9**, 228–239 (2016)
17. Goh, C.J., Yang, X.Q.: Duality in Optimization and Variational Inequalities. Taylor and Francis, London (2002)
18. Ponstein, J.: Seven types of convexity. SIAM Rev. **9**, 115–119 (1967)
19. Bento, G.C., Cruz Neto, J.X., Oliveira, P.R., Soubeyran, A.: The self regulation problem as an inexact steepest descent method for multicriteria optimization. Eur. J. Oper. Res. **235**, 494–502 (2014)
20. Mangasarian, O.L.: Nonlinear Programming. SIAM, Philadelphia (1994)
21. Bento, G.C., Cruz Neto, J.X., Santos, P.S.M.: An inexact steepest descent method for multicriteria optimization on riemannian manifolds. J. Optim. Theory Appl. **159**, 108–124 (2013)
22. Durea, M., Strugariu, R., Tammer, C.: Scalarization in geometric and functional vector optimization revisited. J. Optim. Theory Appl. **159**, 635–655 (2013)
23. Huard, P.: Convergence of the reduced gradient method. In: Mangasarian, O.L., Meyer, R.R., Robinson, S.M. (eds.) Nonlinear Programming, pp. 29–59. Academic Press, New York (1975)
24. Armijo, L.: Minimization of functions having Lipschitz continuous first partial derivatives. Pac. J. Math. **16**, 1–3 (1966)
25. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: empirical results. Evol. Comput. **8**, 173–195 (2000)
26. Bolintinéanu, S., (new name: Bonnel, H.), El Maghri, M.: Pénalisation dans l'optimisation sur l'ensemble faiblement efficient. RAIRO Oper. Res. **31**, 295–310 (1997)
27. Bolintinéanu, S., El Maghri, M.: Second-order efficiency conditions and sensitivity of efficient points. J. Optim. Theory Appl. **98**, 569–592 (1998)
28. El Maghri, M., Bernoussi, B.: Pareto optimizing and Kuhn–Tucker stationary sequences. Numer. Funct. Anal. Optim. **28**, 287–305 (2007)