

On the convergence of steepest descent methods for multiobjective optimization

G. Cocchi, G. Liuzzi, S. Lucidi & M. Sciandrone

Computational Optimization and Applications

An International Journal

ISSN 0926-6003

Volume 77

Number 1

Comput Optim Appl (2020) 77:1-27

DOI 10.1007/s10589-020-00192-0

Your article is protected by copyright and all rights are held exclusively by Springer Science+Business Media, LLC, part of Springer Nature. This e-offprint is for personal use only and shall not be self-archived in electronic repositories. If you wish to self-archive your article, please use the accepted manuscript version for posting on your own website. You may further deposit the accepted manuscript version in any repository, provided it is only made publicly available 12 months after official publication or later and provided acknowledgement is given to the original source of publication and a link is inserted to the published article on Springer's website. The link must be accompanied by the following text: "The final publication is available at link.springer.com".



On the convergence of steepest descent methods for multiobjective optimization

G. Cocchi¹ · G. Liuzzi² · S. Lucidi³ · M. Sciandrone¹

Received: 25 July 2019 / Published online: 5 May 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

In this paper we consider the classical unconstrained nonlinear multiobjective optimization problem. For such a problem, it is particularly interesting to compute as many points as possible in an effort to approximate the so-called Pareto front. Consequently, to solve the problem we define an “a posteriori” algorithm whose generic iterate is represented by a set of points rather than by a single one. The proposed algorithm takes advantage of a linesearch with extrapolation along steepest descent directions with respect to (possibly not all of) the objective functions. The sequence of sets of points produced by the algorithm defines a set of “linked” sequences of points. We show that each linked sequence admits at least one limit point (not necessarily distinct from those obtained by other sequences) and that every limit point is Pareto-stationary. We also report numerical results on a collection of multiobjective problems that show efficiency of the proposed approach over more classical ones.

Keywords Multiobjective optimization · A posteriori method · Steepest descent algorithm

Mathematics Subject Classification 90C30 · 90C56 · 65K05

✉ G. Liuzzi
giampaolo.liuzzi@iasi.cnr.it

G. Cocchi
guido.cocchi@unifi.it

S. Lucidi
lucidi@diag.uniroma1.it

M. Sciandrone
marco.sciandrone@unifi.it

¹ Dipartimento di Ingegneria dell’Informazione, Università di Firenze, Florence, Italy

² Consiglio Nazionale delle Ricerche, Istituto di Analisi dei Sistemi e Informatica, Rome, Italy

³ Dipartimento di Ingegneria Informatica Automatica e Gestionale, Università di Roma “Sapienza”, Rome, Italy

1 Introduction

We consider the following unconstrained multiobjective optimization problem:

$$\min_{x \in \mathbb{R}^n} f_1(x), \dots, f_m(x), \quad (1)$$

where the functions $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, for $i = 1, \dots, m$ are continuously differentiable. Many competing objective functions have to be minimized simultaneously and, to this aim, we need to employ the well-known concept of Pareto optimality. A point is Pareto optimal if there does not exist another point with not greater objective function values and such that there is a strict decrease in at least one objective function value.

One of the most important classes of algorithms for multiobjective optimization is represented by the so-called scalarization methods (see, e.g. [8, 10, 15, 18]), where candidate Pareto solutions of problem (1) are computed by solving one or several parametrized single-objective optimization problems. The main drawback of scalarization methods is the need of defining suitable parameters to obtain a “good” scalarization function, and this requires an insight on the problem structure which might not be available in general.

Here we focus on a class of parameter-free multiobjective optimization algorithms, recently studied, that extends classic optimization methods to multiobjective (both constrained and unconstrained) optimization problems [9, 11, 12, 19]. The common strategy of these methods, that do not exploit a priori information on the objectives, is to compute, starting from the current point, a feasible and descent (for all the objectives) direction and to perform a linesearch along it that produces a new point where a strict decrease of all the objectives is obtained. The computation of the search direction usually involves the solution of a convex quadratic programming problem, while the linesearch is the natural extension to the multiobjective case of the standard backtracking strategy.

These descent methods, similarly to standard algorithms for single-objective optimization, generate a sequence of points and exhibit convergence properties. The convergence analysis shows that, under suitable assumptions, the sequence generated by a descent algorithm admits limit points and each (or at least one) limit point is a Pareto-stationarity point. However, the main aim of a multiobjective optimization algorithm consists in obtaining an approximation to the Pareto front, so that it could be useful to define descent algorithms generating a sequence of sets of non-dominated points with guaranteed convergence properties. Up to our knowledge the only descent algorithm presenting these features has been presented in [13] for constrained multiobjective optimization. The proposed algorithm is based on two main stages: the first one is aimed to enrich the set of non-dominated points using separately each individual objective function, and can be viewed as a heuristic to spread feasible points along the Pareto front; the second one is designed to drive each non-dominated point towards optimality.

Other multiobjective algorithms for generating an approximation of the whole Pareto-front are natural extension of well-known global optimization techniques

(see e.g. [5, 6, 14]). Derivative-free multiobjective optimization algorithms for retrieving an approximation of the Pareto front are proposed, e.g., in [3, 4, 17].

In this work we present a different approach and we propose a steepest descent algorithm for unconstrained multiobjective optimization that produces a set of non-dominated points, i.e. an approximation of the Pareto front for problem (1). The key ingredients that characterize the proposed approach are the following: (i) not all the objective functions are necessarily considered to define the search direction; (ii) a sequence of sets of non-dominated points (rather than a sequence of points) is produced at each iteration; (iii) an Armijo-type linesearch (taking into account all the elements of the current set of non-dominated points) is suitably designed to couple with (i) and (ii). The convergence analysis is based on the concept of linked sequences introduced in [17] and leads to define the theoretical properties of the sequence of sets of non-dominated points produced by the algorithm. We remark that a typical output of standard descent algorithms is a single non-dominated point, while our proposed algorithm generates a sequence of sets of points trying to approximate the Pareto front. This makes the convergence analysis both non-trivial and non-standard. The results of some computational experiments show the validity of the proposed framework.

The paper is organized as follows. In Sect. 2 we introduce some definitions and we report some preliminaries about multiobjective optimization. We also introduce the standard steepest descent method. The proposed steepest descent algorithm that, in contrast to the standard descent approach, produces a set (instead of a single point) of non-dominated points, is defined in Sect. 3. The algorithm is based on a suitable backtracking linesearch whose correctness is proved in the same section. The theoretical properties of the framework are stated and proved in Sect. 4. A different linesearch, where the initial stepsize can vary at every iteration and an extrapolation phase can be performed along the search direction, is presented in Sect. 5 together with the related convergence analysis. The results of computational experiments are shown in Sect. 6. Finally, Sect. 7 contains some concluding remarks.

2 Preliminaries on multiobjective optimization

With reference to Problem (1), we denote by $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ the vector-valued function defined by

$$F(x) = (f_1(x), f_2(x), \dots, f_m(x))^T,$$

and by $J : \mathbb{R}^n \rightarrow \mathbb{R}^{m \times n}$ its Jacobian matrix function, i.e.,

$$J(x) = (\nabla f_1(x) \ \nabla f_2(x) \ \dots \ \nabla f_m(x))^T.$$

Given any two vectors $u, v \in \mathbb{R}^m$,

$$\begin{aligned} u < v &\Leftrightarrow u_i < v_i, \text{ for all } i = 1, \dots, m \\ u \leq v &\Leftrightarrow u_i \leq v_i, \text{ for all } i = 1, \dots, m \\ u \leq v &\Leftrightarrow u \leq v \text{ and } u \neq v. \end{aligned}$$

We can state the following definition of (Pareto) dominance.

Definition 1 (*Pareto dominance*) Given two vectors $x, y \in \mathbb{R}^n$, we say that x (strictly) Pareto dominates y when

$$F(x) \leq F(y) \quad (F(x) \leq F(y)).$$

Then, with reference to problem (1), an ideal solution would be a point $x^* \in \mathbb{R}^n$ that Pareto dominates each other feasible point, i.e.,

$$F(x^*) \leq F(x), \quad \text{for all } x \in \mathbb{R}^n.$$

Unfortunately, such a point x^* very seldom exists, which is why the following definitions of optimality are introduced for multiobjective problems.

Definition 2 (*Weak Pareto optimality*) A point $x^* \in \mathbb{R}^n$ is a weak Pareto optimal point for Problem (1), if there does not exist any $x \in \mathbb{R}^n$ such that

$$F(x) < F(x^*).$$

Definition 3 (*Pareto optimality*) A point $x^* \in \mathbb{R}^n$ is a Pareto optimal point for Problem (1), if there does not exist any $x \in \mathbb{R}^n$ such that

$$F(x) \leq F(x^*).$$

By means of these two definitions, we are able to identify a set of non-dominated points (the so-called Pareto front or frontier) which is constituted by the “optimal” solutions of the multiobjective problem (1).

Finally, we state the definition of *Pareto-stationarity*.

Definition 4 (*Pareto stationarity*) A point $x^* \in \mathbb{R}^n$ is Pareto-stationary for Problem (1) if, for all $d \in \mathbb{R}^n$, an index $j \in \{1, \dots, m\}$ exists such that

$$\nabla f_j(x^*)^\top d \geq 0.$$

If $x \in \mathbb{R}^n$ is not a Pareto-stationary point, then there exists a descent direction v for all the objective functions $f_i, i \in \{1, \dots, m\}$ at x . Given x , let $g_x : \mathbb{R}^n \rightarrow \mathbb{R}$ be the function defined as follows

$$g_x(v) = \max_{i \in \{1, \dots, m\}} \nabla f_i(x)^\top v. \tag{2}$$

Note that g_x is continuous, piecewise linear, and convex. Let us consider the following optimization problem:

$$\min_{v \in \mathbb{R}^n} g_x(v) + \frac{1}{2} \|v\|^2 \tag{3}$$

We observe that the above problem is well-defined, i.e., it admits a unique solution, since the objective function is the maximum of m coercive and strictly

convex functions, so that, it is coercive and stricly convex. Now let $v : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $\theta : \mathbb{R}^n \rightarrow \mathbb{R}$ be the functions such that

$$v(x) = \underset{v \in \mathbb{R}^n}{\operatorname{argmin}} \ g_x(v) + \frac{1}{2} \|v\|^2, \tag{4}$$

$$\theta(x) = \min_{v \in \mathbb{R}^n} g_x(v) + \frac{1}{2} \|v\|^2 = g_x(v(x)) + \frac{1}{2} \|v(x)\|^2. \tag{5}$$

From [12] we get the following result.

Proposition 1 *Let $v : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $\theta : \mathbb{R}^n \rightarrow \mathbb{R}$ be the functions defined as in (4) and (5) respectively. Then the following statements hold:*

- (i) $\theta(x) \leq 0$ for all $x \in \mathbb{R}^n$;
- (ii) x^* is Pareto-stationary if and only if $\theta(x^*) = 0$;
- (iii) the mappings $x \rightarrow v(x)$ and $x \rightarrow \theta(x)$ are continuous.

Finally, we briefly recall the *standard* steepest descent method for unconstrained multiobjective optimization (see, e.g., [12]). For every iteration k we compute the multiobjective steepest descent direction solving the following optimization problem:

$$\min_{v \in \mathbb{R}^n} \max_{i \in \{1, \dots, m\}} \nabla f_i(x_k)^\top v + \frac{1}{2} \|v\|^2. \tag{6}$$

Let us call v_k the solution of problem (6). If $v_k \neq \mathbf{0}$, then we can use it with an Armijo-type linesearch technique to look for a new solution which dominates x_k . Then, we update the current point as

$$x_{k+1} = x_k + \alpha_k v_k$$

Algorithm 1: multiobjective_steepest_descent_algorithm

Data: $x_0 \in \mathbb{R}^n$
for $k = 0, 1, \dots$, **do**
 $v_k \leftarrow \underset{v \in \mathbb{R}^n}{\operatorname{argmin}} \max_{i \in \{1, \dots, m\}} \nabla f_i(x_k)^\top v + \frac{1}{2} \|v\|^2$
 if $v_k = \mathbf{0}$ **then**
 | **return** stationary point x_k
 end
 $\alpha_k \leftarrow \operatorname{armijo_line_search}(x_k, v_k, J(x_k))$
 $x_{k+1} \leftarrow x_k + \alpha_k v_k$
end

Here we describe a multiobjective adaptation for an Armijo-type linesearch:

Algorithm 2: `armijo_line_search`

Data: $x_k \in \mathbb{R}^n, v_k \in \mathbb{R}^n, J(x_k), \gamma \in (0, 1), \delta \in (0, 1), \Delta > 0$
 $\alpha \leftarrow \Delta$
while $F(x_k + \alpha_k v_k) \not\leq F(x_k) + \gamma \alpha J(x_k) v_k$ **do**
 $\alpha \leftarrow \alpha \delta$
end
 $\alpha_k = \alpha$

Remark 1 In Algorithm 2, the computed stepsize α_k is such that

$$F(x_k + \alpha_k v_k) \leq F(x_k) + \gamma \alpha_k J(x_k) v_k,$$

i.e. the point $x_k + \alpha_k v_k$ is such that $F(x_k + \alpha_k v_k)$ strictly (or sufficiently) dominates the corresponding vector of reference values $F(x_k)$.

The following global convergence result holds [12].

Proposition 2 *Let $\{x_k\}$ be the sequence generated by the algorithm. If F has bounded level sets in the sense that $\{x \in \mathbb{R}^n : F(x) \leq F(x_0)\}$ is compact, then each limit point of $\{x_k\}$ is a Pareto-stationary point.*

Newton and Quasi-Newton methods for multiobjective optimization which, similarly to the steepest descent algorithm, are based on the employment of a joint descent direction, have been proposed in several papers (see, e.g., [11, 19])

To our knowledge, steepest descent algorithms and their extensions proposed in the literature are guaranteed to converge to a single non-dominated solution. In the sequel, we present an approach leading to define an algorithm that theoretically converges to a set of non-dominated points.

3 An algorithm for the computation of a set of non-dominated points

In this section, we are interested in the definition of an algorithm that produces a set of non-dominated points, i.e. an approximation of the Pareto front for the multiobjective problem (1).

The key ingredients that characterize the proposed approach are the following:

- (i) not all the objective functions are necessarily considered to define the search direction;
- (ii) a sequence of sets of points (rather than a sequence of points) is produced;
- (iii) an Armijo-type linesearch is suitably defined to cope with (i) and (ii).

With respect to standard steepest descent methods, point (i) allows to compute more easily the possible search directions. In particular, at every iteration k , for a given subset of objective function indices $I \subseteq \{1, \dots, m\}$ and for a (non-dominated) point x_c , it is possible to define

$$v'_k(x_c) := \operatorname{argmin}_{v \in \mathbb{R}^n} \left\{ \max_{i \in I} \nabla f_i(x_c)^\top v + \frac{1}{2} \|v\|^2 \right\}, \tag{7}$$

$$\theta'_k(x_c) := \max_{i \in I} \nabla f_i(x_c)^\top v'_k(x_c) + \frac{1}{2} \|v'_k(x_c)\|^2, \tag{8}$$

$$F_I(x) = |I|\text{-dimensional vector with components } f_i(x), i \in I. \tag{9}$$

In particular, by setting $I = \{1, \dots, m\}$, we have

$$\theta'_k(x_c) = \theta_k(x_c) = \min_{v \in \mathbb{R}^n} \left\{ \max_{i \in \{1, \dots, m\}} \nabla f_i(x_k)^\top v + \frac{1}{2} \|v\|^2 \right\}.$$

We note that the possibility to choose subsets of indices $I \subset \{1, \dots, m\}$ may introduce sort of a controlled randomness in the optimization process which could help to obtain a better approximation of the Pareto front.

Point (ii) is connected with the idea of trying to approximate the Pareto front instead of a single Pareto point. To this aim, the algorithm produces a sequence of sets of points (rather than a sequence of points as usual in the single-objective case) $\{L_k\}$. More in particular, for each iteration k , we indicate by

$$L_k = \{x_h \in \mathbb{R}^n, h = 1, \dots, r_k\}$$

the current finite set of non-dominated solutions computed by the algorithm. For simplicity, in the definition of L_k we omit the dependence on k of every point $x_h \in L_k$.

Finally, concerning point (iii) on the computation of the stepsize, as in the case of single objective optimization where the linesearch must produce a stepsize satisfying a condition of sufficient decreasing, i.e., the stepsize must yield a “sufficiently good” point, here the Armijo-type linesearch determines a stepsize such that the new point is “sufficiently non-dominated” by any other point in the current list $\{\tilde{L}_k\}$. This means that, given the current point x_c , the subset of indices I , the search direction $v'_k(x_c)$, the tentative stepsize α , if there exists a point $x_j \in \tilde{L}_k$ such that

$$F_I(x_j) + \mathbf{1}\gamma\alpha\theta'_k(x_c) < F_I(x_c + \alpha v'_k(x_c))$$

then the stepsize must be reduced.

Definition 5 (Non-dominated solutions with respect to I) $L^I \subseteq \mathbb{R}^n$ is a non-dominated set of points if, for all $x \in L^I$, there does not exist any $y \in L^I$ such that $F_I(y) \leq F_I(x)$.

In this context, given two points $x, y \in \mathbb{R}^n$, a subset of indices $I \subseteq \{1, \dots, m\}$ and a positive constant $\delta > 0$, we say that x is “not weakly better” than y when

$$F_I(x) - \mathbf{1}\delta \not\prec F_I(y),$$

i.e. when an index $i \in I$ exists such that $f_i(y) \leq f_i(x) - \delta$, that is, $f_i(y)$ sufficiently improves w.r.t. $f_i(x)$.

Regarding function $\theta'_k(x_c)$, it is possible to state the following result.

Proposition 3 *If x_c is not a Pareto Stationary point then for any $I \subseteq \{1, \dots, m\}$ we have*

$$\theta_k^I(x_c) < 0.$$

Proof For any $v \in \mathbb{R}^n$ we can write

$$\max_{i \in I} \left\{ \nabla f_i(x_c)^\top v + \frac{1}{2} \|v\|^2 \right\} \leq \max_{i \in \{1, \dots, m\}} \left\{ \nabla f_i(x_c)^\top v + \frac{1}{2} \|v\|^2 \right\},$$

so that we have

$$\theta_k^I(x_c) \leq \theta(x_c) < 0,$$

where the strict inequality holds since x_k is not a Pareto Stationary point. □

Algorithm 3: front_steepest_descent_algorithm

input: an initial set of non-dominated points L_0 .

output: a final list of non-dominated points L^* .

$k \leftarrow 0$

while stopping criterion not satisfied **do**

$\tilde{L}_k \leftarrow L_k$

for $c = 1, \dots, r_k$ **do**

if $x_c \in \tilde{L}_k$ and $\exists I_k^c \subseteq \{1, \dots, m\}$ such that

- $x_c \in \tilde{L}_k^{I_k^c}$, where $\tilde{L}_k^{I_k^c} \subseteq \tilde{L}_k$ is the set of non-dominated points w.r.t I_k^c (see Definition 5)

- $\theta_k^{I_k^c}(x_c) < 0$ **then**

Let $v_k^{I_k^c}(x_c)$ be the direction associated with $\theta_k^{I_k^c}(x_c)$

$\alpha \leftarrow \text{armijo_type_line_search}(x_c, v_k^{I_k^c}(x_c), \theta_k^{I_k^c}(x_c), \tilde{L}_k^{I_k^c})$

$\tilde{L}_k \leftarrow \{x_j \in \tilde{L}_k \mid F(x_c + \alpha v_k^{I_k^c}(x_c)) \not\leq F(x_j)\} \cup \{x_c + \alpha v_k^{I_k^c}(x_c)\}$

else

if $x_c \in \tilde{L}_k$ (i.e. x_c has not been filtered out by previous inner iterations) **then**

x_c is a Pareto Stationary point.

end

end

end

$L_{k+1} \leftarrow \tilde{L}_k$

$k \leftarrow k + 1$

end

$L^* \leftarrow L_k$

return L^*

Algorithm 4: armijo_type_line_search

input: $x_c \in \tilde{L}_k^{I_c^c}, v_k^{I_c^c}(x_c), \theta_k^{I_c^c}(x_c), \tilde{L}_k^{I_c^c}, \Delta > 0, \delta \in (0, 1)$
output: an optimal stepsize α
 Set $\alpha \leftarrow \Delta$
while $\exists x_j \in \tilde{L}_k^{I_c^c}$ s.t. $F_{I_k^c}(x_j) + \mathbf{1}\gamma\alpha\theta_k^{I_c^c}(x_c) < F_{I_k^c}(x_c + \alpha v_k^{I_c^c}(x_c))$ **do**
 | Set $\alpha \leftarrow \delta\alpha$
end
return α

Remark 2 In Algorithm 4, the computed stepsize α is such that any other point $x_j \in \tilde{L}_k^{I_c^c}$ is “not weakly better” than $x_c + \alpha v_k^{I_c^c}(x_c)$, with respect to I_k^c . Namely,

$$\forall x_j \in \tilde{L}_k^{I_c^c}, F_{I_k^c}(x_j) + \mathbf{1}\gamma\alpha\theta_k^{I_c^c}(x_c) \not\prec F_{I_k^c}(x_c + \alpha v_k^{I_c^c}(x_c)),$$

i.e. for every $x_j \in \tilde{L}_k^{I_c^c}$, an index $h_j \in I_k^c$ exists such that

$$f_{h_j}(x_c + \alpha v_k^{I_c^c}(x_c)) \leq f_{h_j}(x_j) + \gamma\alpha\theta_k^{I_c^c}(x_c).$$

Remark 3 We note that, when $m = 1$ (i.e. only one objective function is present) and $L_k = \{x_k\}$, for all k , both Algorithms 2 and 4 boil down to the standard Armijo inexact linesearch. Consequently, both Algorithms 1 and 3 become the standard steepest descent algorithm for single objective optimization.

Instead, if $m > 1$ and $L_k = \{x_k\}$, for all k , then Algorithm 4 tends to produce a bigger stepsize α_k since it must satisfies the weaker requirement

$$F_{I_k}(x_k) + \mathbf{1}\gamma\alpha_k\theta_k^{I_k}(x_k) \not\prec F_{I_k}(x_k + \alpha_k v_k),$$

namely that an index $s \in I_k$ exists such that

$$f_s(x_k + \alpha_k v_k) \leq f_s(x_k) + \gamma\alpha_k\theta_k^{I_k}(x_k). \tag{10}$$

This condition appears to be the weakest extension of the Armjio strategy to the multiobjective case. We recall that in Algorithm 2, inequality (10) must hold for all $s \in I_k$.

Remark 4 Algorithm 3 points out that there is a strict connection between the properties required to the search direction v_c and the possibility of improving a point x_c by using Algorithm 4 along such direction. More precisely, if the current point x_c and the selection of the index set $I \subseteq \{1, \dots, m\}$ were uncoupled, even though we could be able to compute a descent direction with respect to the objective functions with indices in I , the points produced by the linesearch could all be unacceptable because they could all be dominated by the points in the current list \tilde{L} . In order to avoid such possibility, we preliminary check whether x_c is non-dominated with respect to $f_i, i \in I$. In this case, a linesearch along a descent direction with respect to the objective function with indices in I , is guaranteed to produce non-dominated points with respect to \tilde{L} .

For example, if an index $s \in \{1, \dots, m\}$ exists such that

$$x_c = \underset{x \in \tilde{L}_k}{\operatorname{argmin}} f_s(x) \tag{11}$$

then every vector v_k which is a suitable descent direction for the function f_s can be fruitfully used in Algorithm 4 starting from x_c .

In the following proposition we show that, given a subset $I_k^c \subseteq \{1, \dots, m\}$, Algorithm 4 is able to add a new non-dominated solution to \tilde{L}_k .

Proposition 4 *Let $I_k^c \subseteq \{1, \dots, m\}$, $x_c \in \tilde{L}_k^{I_k^c}$ be such that $\theta_k^{I_k^c}(x_c) < 0$, i.e. $v_k^{I_k^c}(x_c)$ exists such that*

$$\nabla f_s(x_c)^\top v_k^{I_k^c}(x_c) + \frac{1}{2} \|v_k^{I_k^c}(x_c)\|^2 < 0$$

$\forall s \in I_k^c$. Then $\exists \alpha > 0$, sufficiently small, such that

$$F_{I_k^c}(x_j) + \mathbf{1}\gamma\alpha\theta_k^{I_k^c}(x_c) \not\leq F_{I_k^c}(x_c + \alpha v_k^{I_k^c}(x_c)), \quad \forall x_j \in \tilde{L}_k^{I_k^c},$$

i.e. the while loop of Algorithm 4 terminates in a finite number of iterations. Furthermore the produced point is not dominated with respect to the set \tilde{L}_k .

Proof First we note that, since $x_c \in \tilde{L}_k^{I_k^c}$, there cannot exist any $x_j \in \tilde{L}_k^{I_k^c}$ such that

$$f_s(x_j) \leq f_s(x_c), \quad \forall s \in I_k^c \tag{12}$$

that is to say that x_j dominates x_c . Now, we proceed by contradiction, and assume that, for all $h = 1, 2, \dots$, an $x_{j_h} \in \tilde{L}_k^{I_k^c}$ exists such that

$$F_{I_k^c}(x_{j_h}) + \mathbf{1}\gamma\delta^h\Delta\theta_k^{I_k^c}(x_c) < F_{I_k^c}(x_c + \delta^h\Delta v_k^{I_k^c}(x_c)).$$

By recalling (12), for all h an index $s_h \in I_k^c$ exists such that:

$$f_{s_h}(x_c) + \gamma\delta^h\Delta\theta_k^{I_k^c}(x_c) < f_{s_h}(x_c + \delta^h\Delta v_k^{I_k^c}(x_c)).$$

Since $|I_k^c|$ is finite, it is possible to consider a subsequence $H \subseteq \{1, 2, \dots\}$ such that $s_h = \bar{s}$, for all $h \in H$. Hence, for all $h \in H$, we would have

$$f_{\bar{s}}(x_c) + \gamma\delta^h\Delta\theta_k^{I_k^c}(x_c) < f_{\bar{s}}(x_c + \delta^h\Delta v_k^{I_k^c}(x_c)).$$

On the other hand, since $v_k^{I_k^c}(x_c)$ is a descent direction for all f_s with $s \in I_k^c$, we have that, for $h \in H$, it must result

$$f_{\bar{s}}(x_c + \delta^h\Delta v_k^{I_k^c}(x_c)) - f_{\bar{s}}(x_c) > \gamma\delta^h\Delta\nabla f_{\bar{s}}(x_c)^\top v_k^{I_k^c}(x_c). \tag{13}$$

By the Mean-value Theorem, we have that

$$f_{\bar{s}}(x_c + \delta^h \Delta v_k^{I_c}(x_c)) - f_{\bar{s}}(x_c) = \delta^h \Delta \nabla f_{\bar{s}}(\xi_h)^\top v_k^{I_c}(x_c) \tag{14}$$

with

$$\xi_h = x_c + t_h \delta^h \Delta v_k^{I_c}(x_c), \quad t_h \in (0, 1).$$

Then, by (13) and (14), we can write

$$\nabla f_{\bar{s}}(\xi_{h_j})^\top v_k^{I_c}(x_c) \geq \gamma \nabla f_{\bar{s}}(x_c)^\top v_k^{I_c}(x_c).$$

Now, by taking the limit for $h \rightarrow \infty, h \in H$, we would contradict that

$v_k^{I_c}(x_c)$ is a descent direction for $f_{\bar{s}}$.

Finally the acceptability criterion implies that the produced point is not dominated. In fact, let $\bar{\alpha}$ denote the stepsize produced by the Armijo-type linesearch then for every $x_j \in \tilde{L}_k^{I_c}$ an index $h_j \in I_k^c$ exists such

$$f_{h_j}(x_j) > f_{h_j}(x_j) + \gamma \bar{\alpha} \theta_k^{I_c}(x_c) \geq f_{h_j}(x_c + \bar{\alpha} v_k^{I_c}(x_c)).$$

Then, by definition of $\tilde{L}_k^{I_c}$, every $\tilde{x}_j \in \tilde{L}_k \setminus \tilde{L}_k^{I_c}$ is dominated by a point $x_j \in \tilde{L}_k^{I_c}$ with respect the functions f_i , for all $i \in I_k^c$, i.e.

$$f_i(\tilde{x}_j) \geq f_i(x_j), \quad \text{for all } i \in I_k^c$$

Therefore, the considered index $h_j \in I_k^c$ is such that

$$f_{h_j}(\tilde{x}_j) \geq f_{h_j}(x_j) > f_{h_j}(x_j) + \gamma \bar{\alpha} \theta_k^{I_c}(x_c) \geq f_{h_j}(x_c + \bar{\alpha} v_k^{I_c}(x_c)).$$

□

4 Convergence analysis

In order to give convergence properties of the algorithm, a definition of *linked sequences* is reported [17].

Definition 6 Let $\{L_k\}$ be the sequence of sets of non-dominated points produced by Algorithm 3. We define a *linked sequence* as a sequence $\{x_{j_k}\}$ such that, for any $k = 1, 2, \dots, x_{j_k} \in L_k$ is generated at iteration $k - 1$ of Algorithm 3-4 by the point $x_{j_{k-1}} \in L_{k-1}$.

Hence, $x_{j_k} = x_{j_{k-1}} + \alpha_{j_k} v_{k-1}^{I_c}(x_{j_{k-1}})$ and it results

$$F_{I_{k-1}}(x_{e_j}) + \mathbf{1} \gamma \alpha_{j_k} \theta_{k-1}^{I_c}(x_{j_{k-1}}) \not\prec F_{I_{k-1}}(x_{j_k})$$

for all $x_{\ell_j} \in L_{k-1}^{k-1} = \{x \in L_{k-1}, \text{ non-dominated with respect to } f_i, i \in I_{k-1}\}$, where $I_{k-1} \subseteq I$ (for simplicity we have not reported the dependence of set I_{k-1} from the point $x_{j_{k-1}}$).

Then we introduce the following assumption needed to prove the convergence results.

Assumption 1 Given a list $L_0 \subset \mathbb{R}^n$ of non-dominated points, a point $x_0 \in L_0$ exists such that:

- (i) x_0 is not Pareto stationary and
- (ii) the set

$$\mathcal{L}(x_0) = \bigcup_{i=1}^m \left\{ x \in \mathbb{R}^n : f_i(x) \leq f_i(x_0) \right\}$$

is compact.

We note that point (ii) of Assumption 1 is stronger than the assumption required to prove convergence of the algorithm in [12], see Proposition 2. However, this is not strange since our acceptance condition in the linesearch (see Algorithm 4) is weaker than that used in Algorithm 2 from [12].

Proposition 5 *Let us assume that Assumption 1 holds. Let $\{L_k\}$ be the sequence of sets of non-dominated points produced by Algorithm 3. Let $\{x_{j_k}\}$ be a linked sequence, then it admits limit points and every limit point is Pareto-stationary for problem (1).*

Proof First of all, we show that the every linked sequence $\{x_{j_k}\}$ admits a limit point \bar{x} . The steps of the Algorithm 3 guarantee that

$$F(x_0) \not\prec F(x_{j_k}), \forall k$$

and that $\forall k$ there exists an index i_{j_k} such that

$$x_{j_k} \in \{x \in \mathbb{R}^n : f_{i_{j_k}}(x) \leq f_{i_{j_k}}(x_0)\}.$$

Therefore we can conclude that

$$x_{j_k} \in \mathcal{L}_0, \forall k.$$

Then Assumption 1 ensures that the linked sequence $\{x_{j_k}\}$ is bounded.

Now, in order to state the main result, we consider a limit point \bar{x} of a linked sequence $\{x_{j_k}\}$, namely a subsequence $\{x_{j_k}\}_K$ such that

$$\lim_{k \rightarrow \infty, k \in K} x_{j_k} = \bar{x}.$$

Now we recall that \bar{x} is Pareto-stationary for problem (1) if and only if $\theta(\bar{x}) = 0$. Assume, by contradiction, \bar{x} is not Pareto-stationary. This is equivalent to say that there exists a scalar $\varepsilon > 0$ such that

$$\theta(x_{j_k}) \leq -\varepsilon < 0, \quad \forall k \in K$$

and

$$\theta_k^{I_k}(x_{j_k}) \leq -\varepsilon, \quad \forall k \in K, \tag{15}$$

Now, we prove that

$$\lim_{k \rightarrow \infty, k \in K} \alpha_{j_{k+1}} \theta_k^{I_k}(x_{j_k}) = 0. \tag{16}$$

In fact, let us assume, by contradiction, that there exists a set $\bar{K} \subseteq K$ such that:

$$-\alpha_{j_{k+1}} \theta_k^{I_k}(x_{j_k}) \geq \delta > 0 \quad \forall k \in \bar{K}. \tag{17}$$

Since $\{x_{j_k}\}$ is a linked sequence, for all $x_{\ell_j} \in L_k^{I_k}$ and $k \in \bar{K}$, it results

$$F_{I_k}(x_{\ell_j}) + \gamma \alpha_{j_{k+1}} \theta_k^{I_k}(x_{j_k}) \not\leq F_{I_k}(x_{j_{k+1}}). \tag{18}$$

By using (17), we obtain that, for all $x_{\ell_j} \in L_k^{I_k}$ and $k \in \bar{K}$,

$$F_{I_k}(x_{\ell_j}) - \gamma \delta \not\leq F_{I_k}(x_{j_{k+1}}). \tag{19}$$

Now, let \tilde{k} be the smallest index such that $\tilde{k} \in \bar{K}$ and $\tilde{k} > k$, then for all $\tilde{x}_{\ell_j} \in L_{\tilde{k}}^{I_{\tilde{k}}}$,

$$F_{I_{\tilde{k}}}(\tilde{x}_{\ell_j}) - \gamma \delta \not\leq F_{I_{\tilde{k}}}(x_{j_{\tilde{k}}}). \tag{20}$$

If $x_{j_k} \in L_{\tilde{k}}^{I_{\tilde{k}}}$, it satisfies (20). On the other hand, if $x_{j_k} \notin L_{\tilde{k}}^{I_{\tilde{k}}}$, it is dominated by a point $\tilde{x}_{\ell_j} \in L_{\tilde{k}}^{I_{\tilde{k}}}$ and, again, it satisfies (20). Therefore an index $s_{\tilde{k}} \in I_{\tilde{k}}$ exists such that

$$f_{s_{\tilde{k}}}(x_{j_k}) - \gamma \delta \geq f_{s_{\tilde{k}}}(x_{j_{\tilde{k}}}). \tag{21}$$

Now, since $s_{\tilde{k}} \in \{1, \dots, m\}$ and $I_{\tilde{k}} \subseteq \{1, \dots, m\}$, a subset $\tilde{K} \subseteq \bar{K}$ exist such that, for all $k \in \tilde{K}$, $s_{\tilde{k}} = \bar{s}$ and $I_{\tilde{k}} = \bar{I}$ and where, again $\tilde{k}(k)$ is the smallest index such that $\tilde{k}(k) \in K$ and $\tilde{k}(k) > k$.

Then, the definitions of \tilde{K} and $\tilde{k}(k)$ imply that

$$\lim_{k \rightarrow \infty, k \in \tilde{K}} x_{j_k} = \bar{x} \quad \lim_{k \rightarrow \infty, k \in \tilde{K}} x_{j_{\tilde{k}(k)}} = \bar{x}.$$

But then (21) points out the contradiction that

$$-\gamma \delta \geq \lim_{k \rightarrow \infty, k \in \tilde{K}} \left[f_{\bar{s}}(x_{j_{\tilde{k}(k)}}) - f_{\bar{s}}(x_{j_k}) \right] = 0.$$

Therefore, in this first part of the proof, we have shown that if the accumulation point \bar{x} of the sequence $\{x_{j_k}\}$ is not a Pareto stationary point then, by using (15) and (16), we have that:

$$\lim_{k \rightarrow \infty, k \in K} \alpha_{j_{k+1}} = 0. \tag{22}$$

This limit implies that, for sufficiently large values of k , we have that $\alpha_{j_{k+1}} < \Delta$, i.e. the condition of the while-loop in Algorithm 4 is eventually satisfied. Then the steps of Algorithm 4 and the definition of L_k imply that there exists $x_{h_{e_j}} \in L_k$ and $k \in K$, such that

$$F_{I_k}(x_{h_{e_j}}) + \gamma \frac{\alpha_{j_{k+1}}}{\delta} \theta_k^{I_k}(x_{j_k}) < F_{I_k}\left(x_{j_k} + \frac{\alpha_{j_{k+1}}}{\delta} v_k^{I_k}(x_{j_k})\right), \tag{23}$$

$$F_{I_k}(x_{h_{e_j}}) \not\leq F_{I_k}(x_{j_k}). \tag{24}$$

Now (23) and (24) yield that an index $s_k \in I_k$ exists such that

$$f_{s_k}(x_{j_k}) + \gamma \frac{\alpha_{j_{k+1}}}{\delta} \theta_k^{I_k}(x_{j_k}) < f_{s_k}\left(x_{j_k} + \frac{\alpha_{j_{k+1}}}{\delta} v_k^{I_k}(x_{j_k})\right)$$

namely

$$f_{s_k}\left(x_{j_k} + \frac{\alpha_{j_{k+1}}}{\delta} v_k^{I_k}(x_{j_k})\right) - f_{s_k}(x_{j_k}) > \gamma \frac{\alpha_{j_{k+1}}}{\delta} \theta_k^{I_k}(x_{j_k}).$$

Now, since $s_k \in I_k$ and $I_k \subseteq \{1, \dots, m\}$, we can consider a subset $\tilde{K} \subseteq K$ such that, for all $k \in \tilde{K}$, $s_k = \bar{s}$ and $I_k = \bar{I}$, so that

$$f_{\bar{s}}\left(x_{j_k} + \frac{\alpha_{j_{k+1}}}{\delta} v_k^{\bar{I}}(x_{j_k})\right) - f_{\bar{s}}(x_{j_k}) > \gamma \frac{\alpha_{j_{k+1}}}{\delta} \theta_k^{\bar{I}}(x_{j_k}).$$

By the Mean-value Theorem, we have that

$$f_{\bar{s}}\left(x_{j_k} + \frac{\alpha_{j_{k+1}}}{\delta} v_k^{\bar{I}}(x_{j_k})\right) - f_{\bar{s}}(x_{j_k}) = \frac{\alpha_{j_{k+1}}}{\delta} \nabla f_{\bar{s}}(\xi_{j_k})^\top v_k^{\bar{I}}(x_{j_k})$$

with

$$\xi_{j_k} = x_{j_k} + t_{j_k} \frac{\alpha_{j_{k+1}}}{\delta} v_k^{\bar{I}}(x_{j_k}), \quad t_{j_k} \in (0, 1).$$

Then, we can write

$$\nabla f_{\bar{s}}(\xi_{j_k})^\top v_k^{\bar{I}}(x_{j_k}) \geq \gamma \theta_k^{\bar{I}}(x_{j_k}),$$

from which we get

$$\nabla f_{\bar{s}}(x_{j_k})^\top v_k^{\bar{j}}(x_{j_k}) + (\nabla f_{\bar{s}}(\xi_{j_k}) - \nabla f_{\bar{s}}(x_{j_k}))^\top v_k^{\bar{j}}(x_{j_k}) \geq \gamma \theta_k^{\bar{j}}(x_{j_k}).$$

The definition of function $\theta_k^{\bar{j}}$ gives

$$\theta_k^{\bar{j}}(x_{j_k}) + (\nabla f_{\bar{s}}(\xi_{j_k}) - \nabla f_{\bar{s}}(x_{j_k}))^\top v_k^{\bar{j}}(x_{j_k}) \geq \gamma \theta_k^{\bar{j}}(x_{j_k})$$

and

$$(1 - \gamma)\theta_k^{\bar{j}}(x_{j_k}) + (\nabla f_{\bar{s}}(\xi_{j_k}) - \nabla f_{\bar{s}}(x_{j_k}))^\top v_k^{\bar{j}}(x_{j_k}) \geq 0.$$

Using (15) we have

$$-(1 - \gamma)\varepsilon + (\nabla f_{\bar{s}}(\xi_{j_k}) - \nabla f_{\bar{s}}(x_{j_k}))^\top v_k^{\bar{j}}(x_{j_k}) \geq 0$$

By taking the limit for $k \rightarrow \infty$ and $k \in \tilde{K}$, recalling that $\alpha_{j_{k+1}} \rightarrow 0$ and considering the boundedness of $v_k^{\bar{j}}(x_{j_k})$, we obtain the contradiction:

$$-(1 - \gamma)\varepsilon \geq 0$$

and this concludes the proof. □

5 The Armijo-type extrapolation technique

In order to improve the ability of Algorithm 3 of spanning the space, we introduce a new Armijo-type linesearch technique (see Algorithm 5). This new linesearch differs from Algorithm 4 for the following reasons:

- (i) the initial stepsize can vary at every iteration;
- (ii) if the initial stepsize satisfies the acceptability criterion, the algorithm performs an extrapolation along the considered direction;
- (iii) the algorithm can produce more than one stepsize, i.e., a set α^* of stepsizes.

In order to satisfy points (i) and (ii), it is possible to draw inspiration from Armijo-type linesearches with extrapolation. When $m = 1$ and $L_k = \{x_k\}$ for all k , these linesearches start from any initial value of the steplength and then updates it to determine a steplength α_k such that:

$$f_1(x_k + \alpha_k v_k) \leq f_1(x_k) + \gamma \alpha_k \nabla f_1(x_k)^\top v_k, \tag{25}$$

$$f_1\left(x_k + \frac{\alpha_k}{\delta} v_k\right) > f_1(x_k) + \gamma \frac{\alpha_k}{\delta} \nabla f_1(x_k)^\top v_k, \tag{26}$$

with $\delta \in (0, 1)$.

When $m > 1$ and $L_k = \{x_k\}$ for all k (where $x_k = x_c$ as used in the Algorithm), Algorithm 4 suggests that condition (25) can be replaced by the requirement that a function $f_s, s \in I_k$ exists such that (25) holds, namely by the condition:

$$F_{I_k}(x_k) + \mathbf{1}\gamma\alpha_k\theta_k^{I_k^c}(x_k) \not\leq F_{I_k}(x_k + \alpha_kv_k). \tag{27}$$

As regards condition (26), its weakest extension is to require that condition (26) holds for all the functions $f_s, s \in I_k$, namely:

$$F_{I_k}(x_k) + \mathbf{1}\gamma\frac{\alpha_k}{\delta}\theta_k^{I_k^c}(x_k) < F\left(x_k + \frac{\alpha_k}{\delta}v_k\right). \tag{28}$$

In Algorithm 5, since $m > 1$ and $L_k \supseteq \{x_k\}$, conditions (27) and (28) are replaced by the following:

$$F_{I_k^c}(x_j) + \mathbf{1}\gamma\alpha_c\theta_k^{I_k^c}(x_k) \not\leq F_{I_k^c}(x_k + \alpha_cv_k^k(x_k)), \quad \forall x_j \in \tilde{L}_k^{I_k^c}, \tag{29}$$

$$F_{I_k^c}(x_j) + \mathbf{1}\gamma\frac{\alpha_c}{\delta}\theta_k^{I_k^c}(x_k) < F_{I_k^c}\left(x_k + \frac{\alpha_c}{\delta}v_k^k(x_k)\right), \quad \exists x_j \in \tilde{L}_k^{I_k^c}. \tag{30}$$

It is worth noting the different roles that an extrapolation technique can play in the case of a single-objective optimization, as opposed to a multi-objective minimization. In the first case, every intermediate trial value of the steplength produces a point where the objective function is better than the starting point but it is worse than the function value obtained in the final value of the steplength (in terms of multi-objective optimization we can say that the intermediate points produced by the extrapolation phase are dominated by the last one). Instead, in a multi-objective minimization, an extrapolation technique may produce intermediate trial points that are not dominated (i) by all the points belonging to the set $\tilde{L}_k^{I_k^c}$ and (ii) by the other points produced by the extrapolation phase. Such intermediate trial points can be very useful in trying to approximate a Pareto curve. For this reason, Algorithm 5 return a set \mathcal{A}^* of steplengths. In fact, in addition to determining the steplength satisfying conditions (29) and (30), Algorithm 5 also returns the steplengths for which a function f_s , with $s \in I_k^c$ exists such that:

$$F_{I_k^c}(x_j) + \mathbf{1}\gamma\alpha\theta_k^{I_k^c}(x_c) \not\leq F_{I_k^c}(x_c + \alpha v_k^k(x_c)), \quad \forall x_j \in \tilde{L}_k^{I_k^c}, \tag{31}$$

$$f_s(x_c + \alpha v_k^k(x_c)) + \gamma\frac{1-\delta}{\delta}\alpha\theta_k^{I_k^c}(x_c) \leq f_s\left(x_c + \frac{\alpha}{\delta}v_k^k(x_c)\right). \tag{32}$$

The particular structure of Eq. (32) is able to guarantee the theoretical properties of the proposed method (see the next section). From the intuitive point of view, if we set

$$\tilde{x}_c = x_c + \alpha v_k^k(x_c), \quad \tilde{\alpha} = (1 - \delta)\alpha$$

we can rewrite (32) in the following way:

$$f_s(\tilde{x}_c) + \gamma\frac{\tilde{\alpha}}{\delta}\theta_k^{I_k^c}(x_c) \leq f_s\left(\tilde{x}_c + \frac{\tilde{\alpha}}{\delta}v_k^k(x_c)\right),$$

which has a structure similar to that of condition (26) for single-objective minimization.

Algorithm 5: `armijo_type_line_search` with extrapolation

input: $x_c \in \tilde{L}_k^{I_c^c}, v_k^{I_c^c}(x_c), \theta_k^{I_c^c}(x_c), \tilde{L}_k, \Delta_k > 0, \delta \in (0, 1), \gamma \in (0, 1)$
output: a set of steps \mathcal{A}^*

```

 $\mathcal{A}^* \leftarrow \emptyset$ 
 $\alpha \leftarrow \Delta_k$ 
if  $\exists x_j \in \tilde{L}_k^{I_c^c}$  s.t.  $F_{I_k^c}(x_j) + \mathbf{1}\gamma\alpha\theta_k^{I_c^c}(x_c) < F_{I_k^c}(x_c + \alpha v_k^{I_c^c}(x_c))$  then
  while  $\exists x_j \in \tilde{L}_k^{I_c^c}$  s.t.  $F_{I_k^c}(x_j) + \mathbf{1}\gamma\alpha\theta_k^{I_c^c}(x_c) < F_{I_k^c}(x_c + \alpha v_k^{I_c^c}(x_c))$  do
     $\alpha \leftarrow \delta\alpha$ 
  end
   $\mathcal{A}^* \leftarrow \{\alpha\}$ 
else
  while  $F_{I_k^c}(x_j) + \mathbf{1}\gamma\frac{\alpha}{\delta}\theta_k^{I_c^c}(x_c) \not< F_{I_k^c}(x_c + \frac{\alpha}{\delta}v_k^{I_c^c}(x_c)), \forall x_j \in \tilde{L}_k^{I_c^c}$  do
    if  $F_{I_k^c}(x_c + \alpha v_k^{I_c^c}(x_c)) + \mathbf{1}\gamma\frac{1-\delta}{\delta}\alpha\theta_k^{I_c^c}(x_c) \not> F_{I_k^c}(x_c + \frac{\alpha}{\delta}v_k^{I_c^c}(x_c))$ 
      then
         $\mathcal{A}^* \leftarrow \mathcal{A}^* \cup \{\alpha\}$ 
      end
     $\alpha \leftarrow \frac{\alpha}{\delta}$ 
  end
end
return  $\mathcal{A}^*$ 

```

The following proposition states that Algorithm 5 is well defined.

Proposition 6 *Let \tilde{L}_k be the current list of non-dominated points, $I_k^c \subseteq \{1, \dots, m\}$ and $x_c \in \tilde{L}_k^{I_c^c}$ be such that $\theta_k^{I_c^c}(x_c) < 0$, i.e. $v_k^{I_c^c}(x_c)$ exists such that*

$$\nabla f_s(x_c)^\top v_k^{I_c^c}(x_c) + \frac{1}{2} \|v_k^{I_c^c}(x_c)\|^2 < 0, \quad \forall s \in I_k^c. \tag{33}$$

Then Algorithm 5 is well defined, namely it cannot infinitely cycle and it produces at least a point which is not dominated with respect to the set \tilde{L}_k .

Proof In case that the first if-instruction of the algorithm is satisfied, the same reasonings used in the proof of Proposition 4 prove that the first while loop of Algorithm 5 cannot infinitely cycle and a point is produced which is not dominated with respect to \tilde{L}_k .

Therefore, by contradiction, we assume that the second while loop of the algorithm infinitely cycles, i.e., a monotonically increasing sequence of positive numbers $\{\alpha_h\}$, where $\alpha_h = \Delta_k / \delta^h, h \rightarrow \infty$, exists such that:

$$F_{I_k^c}(x_j) + \mathbf{1}\gamma \frac{\alpha_h}{\delta} \theta_k^{I_k^c}(x_c) \not\prec F_{I_k^c}\left(x_c + \frac{\alpha_h}{\delta} v_k^{I_k^c}(x_c)\right), \quad \forall x_j \in \tilde{L}_k^{I_k^c},$$

and, in particular if $x_j = x_c$,

$$F_{I_k^c}(x_c) + \mathbf{1}\gamma \frac{\alpha_h}{\delta} \theta_k^{I_k^c}(x_c) \not\prec F_{I_k^c}\left(x_c + \frac{\alpha_h}{\delta} v_k^{I_k^c}(x_c)\right). \tag{34}$$

The previous (34) implies that, for all h , an index $s_h \in I_k^c$ exists such that

$$f_{s_h}(x_c) + \gamma \frac{\alpha_h}{\delta} \theta_k^{I_k^c}(x_c) \geq f_{s_h}\left(x_c + \frac{\alpha_h}{\delta} v_k^{I_k^c}(x_c)\right).$$

Recalling, again, that $s_h \in \{1, \dots, m\}$ and that $I_k^c = \bar{I}$, we can consider a subset K , such that, for all $h \in K$, it results $s_h = \bar{s}$, so that

$$f_{\bar{s}}(x_c) + \gamma \frac{\alpha_h}{\delta} \theta_k^{\bar{I}}(x_c) \geq f_{\bar{s}}\left(x_c + \frac{\alpha_h}{\delta} v_k^{\bar{I}}(x_c)\right).$$

This relation and the fact that $\alpha_h \rightarrow \infty$ contradict Assumption 1.

Now we prove that, during the second while loop, the set \mathcal{A}^* is updated at least once.

Since the second while loop terminates in a finite number of steps, an index \bar{h} , a step $\alpha_{\bar{h}}$ and a point $\tilde{x}_j \in \tilde{L}_k^{I_k^c}$ exist such that

$$F_{I_k^c}(\tilde{x}_j) + \mathbf{1}\gamma \frac{\alpha_{\bar{h}}}{\delta^{\bar{h}}} \theta_k^{I_k^c}(x_c) < F_{I_k^c}\left(x_c + \frac{\alpha_{\bar{h}}}{\delta^{\bar{h}}} v_k^{I_k^c}(x_c)\right). \tag{35}$$

Namely, the test of the while loop is not satisfied. Instead, at the previous step, the test is verified:

$$F_{I_k^c}(x_j) + \mathbf{1}\gamma \frac{\alpha_{\bar{h}}}{\delta^{\bar{h}-1}} \theta_k^{I_k^c}(x_c) \not\prec F_{I_k^c}\left(x_c + \frac{\alpha_{\bar{h}}}{\delta^{\bar{h}-1}} v_k^{I_k^c}(x_c)\right), \quad \forall x_j \in \tilde{L}_k^{I_k^c}.$$

This means that for $\tilde{x}_j \in \tilde{L}_k^{I_k^c}$ an index $s_j \in I_k^c$ exists such that:

$$f_{s_j}(\tilde{x}_j) + \gamma \frac{\alpha_{\bar{h}}}{\delta^{\bar{h}-1}} \theta_k^{I_k^c}(x_c) \geq f_{s_j}\left(x_c + \frac{\alpha_{\bar{h}}}{\delta^{\bar{h}-1}} v_k^{I_k^c}(x_c)\right). \tag{36}$$

On the other hand (35) yields:

$$f_{s_j}(\tilde{x}_j) + \gamma \frac{\alpha_{\bar{h}}}{\delta^{\bar{h}}} \theta_k^{I_k^c}(x_c) < f_{s_j}\left(x_c + \frac{\alpha_{\bar{h}}}{\delta^{\bar{h}}} v_k^{I_k^c}(x_c)\right). \tag{37}$$

By combining (36) and (37) it is possible to obtain:

$$f_{s_j}\left(x_c + \frac{\alpha_{\bar{h}}}{\delta^{\bar{h}-1}} v_k^{I_k^c}(x_c)\right) + \gamma(1 - \delta) \frac{\alpha_{\bar{h}}}{\delta^{\bar{h}}} \theta_k^{I_k^c}(x_c) < f_{s_j}\left(x_c + \frac{\alpha_{\bar{h}}}{\delta^{\bar{h}}} v_k^{I_k^c}(x_c)\right).$$

This inequality show that the if-condition in the second while loop is satisfied (by setting $\alpha = \frac{\alpha_{\bar{h}}}{\delta^{\bar{h}-1}}$) and hence the set \mathcal{A}^* is updated. Then, the same arguments of the

proof of Proposition 6 imply that it is produced a point which is not dominated with respect to L_k . \square

Since Algorithm 5 generates a sequence of steps along the direction $v_k^J(x_c)$, in order to correctly update the list \tilde{L}_k , line 10 of Algorithm 3 have to be changed as follows:

```

for  $\alpha \in \mathcal{A}^*$  do
     $\tilde{L}_k \leftarrow \{x_j \in \tilde{L}_k \mid F(x_c + \alpha v_k^J(x_c)) \not\leq F(x_j)\} \cup \{x_c + \alpha v_k^J(x_c)\}$ 
end
    
```

Finally we prove the global convergence properties of Algorithm 3 where Algorithm 5 is used in place of Algorithm 4, which we denote Algorithm 3-5 in the following.

Proposition 7 *Let us assume that Assumption 1 holds. Let $\{L_k\}$ be the sequence of sets of non-dominated points produced by the Algorithms 3–5. Let $\{x_{j_k}\}$ be a linked sequence, then it admits limit points and every limit point is Pareto-stationary for problem (1).*

Proof By using Assumption 1 and by repeating the same reasoning of the first part of the proof of Proposition 5, it is possible to prove that the sequence $\{x_{j_k}\}$ is bounded.

Now, let \bar{x} be a limit point of a linked sequence $\{x_{j_k}\}$ and let $\{x_{j_k}\}_K$ be a subsequence such that

$$\lim_{k \rightarrow \infty, k \in K} x_{j_k} = \bar{x}.$$

Assume, by contradiction, that \bar{x} is not Pareto-stationary and that there exist a scalar $\varepsilon > 0$ such that

$$\theta(x_{j_k}) \leq -\varepsilon < 0, \quad \forall k \in K$$

and

$$\theta_k^{I_k}(x_{j_k}) \leq -\varepsilon, \quad \forall k \in K, \tag{38}$$

Now we note that the steps of Algorithm 5 ensure that also the points of every linked sequence $\{x_{j_k}\}$ produced by Algorithm 3-5 satisfy the property that, for all $x_{\ell_j} \in L_k$

$$F_{I_k}(x_{\ell_j}) + \gamma \alpha_{j_{k+1}} \theta_k^{I_k}(x_{j_k}) \not\leq F_{I_k}(x_{j_{k+1}}), \tag{39}$$

therefore it possible to use the same arguments of the proof of Proposition 5 and to ensure that

$$\lim_{k \rightarrow \infty, k \in K} \alpha_{j_{k+1}} \theta_k^{I_k}(x_{j_k}) = 0, \tag{40}$$

which, by using (38), yields that

$$\lim_{k \rightarrow \infty, k \in K} \alpha_{j_{k+1}} = 0. \tag{41}$$

The same reasoning of the proof of Proposition 5 proves the thesis of the proposition in case that a subsequence $\{x_{j_k}\}_{\bar{K}}$, with $\bar{K} \subseteq K$, where the points x_{j_k} are produced by if-test in the first while-loop of Algorithm 5.

Now we consider the case that the points of the subsequence $\{x_{j_k}\}_{\hat{K}}$, with $\hat{K} \subseteq K$, are produced by the if-test in the second while-loop. This implies that for all $k \in \hat{K}$

$$F_{I_k} \left(x_{j_k} + \frac{\alpha_{j_{k+1}}}{\delta} v_k^{I_k}(x_{j_k}) \right) \not\leq F_{I_k}(x_{j_k} + \alpha_{j_{k+1}} v_k^{I_k}(x_{j_k})) + \mathbf{1}\gamma \frac{1-\delta}{\delta} \alpha_{j_{k+1}} \theta_k^{I_k}(x_{j_k})$$

which, by setting $\tilde{x}_{j_k} = x_{j_k} + \alpha_{j_{k+1}} v_k^{I_k}(x_{j_k})$, gives

$$F_{I_k} \left(\tilde{x}_{j_k} + \frac{1-\delta}{\delta} \alpha_{j_{k+1}} v_k^{I_k}(x_{j_k}) \right) \not\leq F_{I_k}(\tilde{x}_{j_k}) + \mathbf{1}\gamma \frac{1-\delta}{\delta} \alpha_{j_{k+1}} \theta_k^{I_k}(x_{j_k})$$

which yields that an index $s_k \in I_k$ exists such that

$$f_{s_k} \left(\tilde{x}_{j_k} + \frac{1-\delta}{\delta} \alpha_{j_{k+1}} v_k^{I_k}(x_{j_k}) \right) \geq f_{s_k}(\tilde{x}_{j_k}) + \gamma \frac{1-\delta}{\delta} \alpha_{j_{k+1}} \theta_k^{I_k}(x_{j_k})$$

namely

$$f_{s_k} \left(\tilde{x}_{j_k} + \frac{1-\delta}{\delta} \alpha_{j_{k+1}} v_k^{I_k}(x_{j_k}) \right) - f_{s_k}(\tilde{x}_{j_k}) \geq \gamma \frac{1-\delta}{\delta} \alpha_{j_{k+1}} \theta_k^{I_k}(x_{j_k}).$$

Recalling that $s_k \in I_k$ and $I_k \subseteq \{1, \dots, m\}$, we can consider a subset $\tilde{K} \subseteq \hat{K}$ such that, for all $k \in \tilde{K}$, $s_k = \bar{s}$ and $I_k = \bar{I}$, so that

$$f_{\bar{s}} \left(\tilde{x}_{j_k} + \frac{1-\delta}{\delta} \alpha_{j_{k+1}} v_k^{\bar{I}}(x_{j_k}) \right) - f_{\bar{s}}(\tilde{x}_{j_k}) \geq \gamma \frac{1-\delta}{\delta} \alpha_{j_{k+1}} \theta_k^{\bar{I}}(x_{j_k})$$

By the Mean-value Theorem, we have that

$$f_{\bar{s}} \left(\tilde{x}_{j_k} + \frac{1-\delta}{\delta} \alpha_{j_{k+1}} v_k^{\bar{I}}(x_{j_k}) \right) - f_{\bar{s}}(\tilde{x}_{j_k}) = \frac{1-\delta}{\delta} \alpha_{j_{k+1}} \nabla f_{\bar{s}}(\xi_{j_k})^\top v_k^{\bar{I}}(x_{j_k})$$

with

$$\xi_{j_k} = \tilde{x}_{j_k} + t_{j_k} \frac{1-\delta}{\delta} \alpha_{j_{k+1}} v_k^{\bar{I}}(x_{j_k}), \quad t_{j_k} \in (0, 1).$$

Then, we can write

$$\nabla f_{\bar{s}}(\xi_{j_k})^\top v_k^{\bar{I}}(x_{j_k}) \geq \gamma \theta_k^{\bar{I}}(x_{j_k}).$$

By using, again, the definition of function $\theta_k^{\bar{I}}$ we have:

$$\theta_k^{\bar{I}}(x_{j_k}) + (\nabla f_{\bar{s}}(\xi_{j_k}) - \nabla f_{\bar{s}}(x_{j_k}))^\top v_k^{\bar{I}}(x_{j_k}) \geq \gamma \theta_k^{\bar{I}}(x_{j_k})$$

and

$$(1 - \gamma)\theta_k^J(x_{j_k}) + (\nabla f_{\bar{s}}(\xi_{j_k}) - \nabla f_{\bar{s}}(x_{j_k}))^\top v_k^J(x_{j_k}) \geq 0.$$

Then (38) implies:

$$-(1 - \gamma)\epsilon + (\nabla f_{\bar{s}}(\xi_{j_k}) - \nabla f_{\bar{s}}(x_{j_k}))^\top v_k^J(x_{j_k}) \geq 0$$

By taking the limit for $k \rightarrow \infty$ and $k \in \tilde{K}$, (41) and the boundedness of $v_k^J(x_{j_k})$, we get the contradiction:

$$-(1 - \gamma)\epsilon \geq 0$$

and this concludes the proof of the proposition. □

6 Preliminary numerical results

In this section, we report some preliminary numerical results, in order to assess the effectiveness of the proposed framework.

The aim is to show how the proposed framework performs with respect to a multistart version of an *a posteriori* steepest descent algorithm (namely, MULTI-START-SD), which iteratively calls Algorithm 1 starting from different randomly sampled points.

Then, considering that several selection techniques of the subset I could be used in Algorithm 3, two variants are proposed and compared. Specifically,

- **Front Steepest Descent** (FRONT-SD): it represents a basic version of the algorithm in which, at every iteration k and for each point $x_c \in L_k$, only the steepest descent direction is computed, i.e. $I = \{1, \dots, m\}$.
- **Front Incremental Steepest Descent** (FRONT-INCREMENTAL-SD): in this version of the algorithm, at every iteration k and for each point $x_c \in L_k$, all the elements of the power set $2^{\{1, \dots, m\}}$ are used to compute search directions.

For what concerns the choice of the linesearch technique, a comparison between the first proposed strategy for dealing with non-dominated solutions, namely STANDARD (see Algorithm 4), and the extrapolation technique, namely EXTRA (see Algorithm 5), is reported.

Test problems: we considered the set of 10 different unconstrained multiobjective problems used for the CEC2009 competition [20], whose dimension n is equal to 30 and with a number m of objectives belonging to the set $\{2, 3\}$. In order to increase the dataset size, we vary the number of variables from 5 to 50 with step 5, obtaining a set of 100 problems.

Implementation details: we implemented Algorithm 3 in Python 3.6 using Tensorflow 1.5 [1] for computing derivatives and Gurobi [16] for solving quadratic programming problems like (789). Additional parameters for the linesearches are given:

$$\Delta_k = 1, \quad \delta = 0.5, \quad \gamma = 10^{-4}.$$

The algorithms stop when one of the two following stopping conditions is met:

- a maximum number of function evaluations is reached (a Jacobian computation costs n).
- all points $x_c \in L_k$ are such that $\theta_k^{I^c}(x_c) \geq 0$ for any subset $I^c \subseteq \{1, \dots, m\}$.

For our experiments, the maximum number of function evaluations is set to 20,000.

Since a box-constrained version for each problem is given in [20], then the centroid of the hyperbox $[\ell, u]$ is chosen as starting point, i.e. $L_0 = \{x_0\}$ with x_0 such that:

$$(x_0)_i = \frac{\ell_i + u_i}{2} \quad \forall i \in \{1, \dots, n\} \tag{42}$$

Since some test problems contain objective functions that are not defined everywhere, infinite values are assigned to the singularities, while points of non-differentiability are not considered for space exploration.

Performance metrics: Purity, Spread Γ and Spread Δ metrics, defined in [4], have been used with the performance profiles benchmarking technique [7] for comparing the performance. We recall that the Purity metric measures the quality of the generated front, i.e. how good the non-dominated points computed by a solver are with respect to those computed by any other solver. Note that, for each problem p , the “reference” Pareto front F_p is calculated by first computing

$$F'_p = \bigcup_s F_{p,s},$$

where $F_{p,s}$ denotes the set of non-dominated solutions found by solver s , and then removing from this set any dominated solution, that is

$$F_p = \{x \in F'_p : \nexists y \in F'_p \text{ s.t. } F(y) \leq F(x)\}. \tag{43}$$

On the other hand, the Spread metrics are essential to measure the uniformity of the generated front in the objectives space. Particularly, Spread Γ measures the maximum “hole” between adjacent points in the objective space, while Spread Δ is quite similar to the standard deviation of the “hole” sizes in the Pareto front.

First of all, we separately compared the two proposed variants with both line-search techniques. The results of this preliminary experimentation showed the effectiveness of the EXTRA variant, in terms of Purity and Spread Γ with respect to the STANDARD one. Given these results, from now onward, the extrapolation technique will always be used for all the other comparisons. In order not to overburden the notation in figure legends, we do not explicitly add the “EXTRA” suffix to algorithm names.

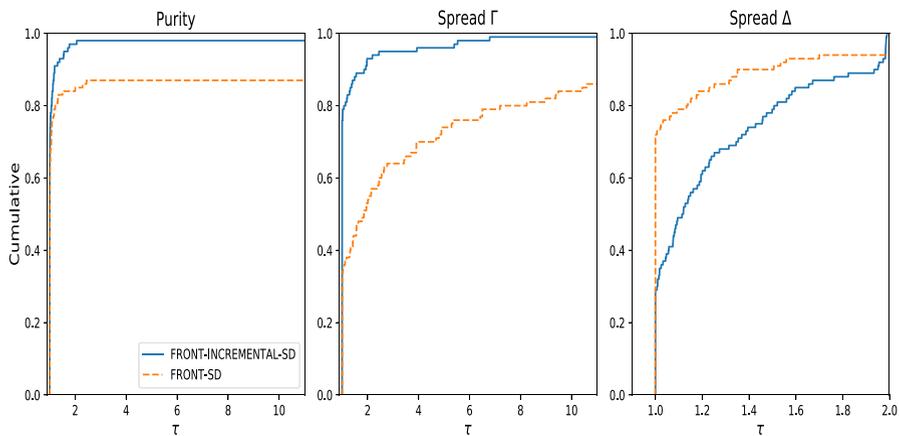


Fig. 1 Performance profiles of Purity and Spread metrics for FRONT-INCREMENTAL-SD and FRONT-SD

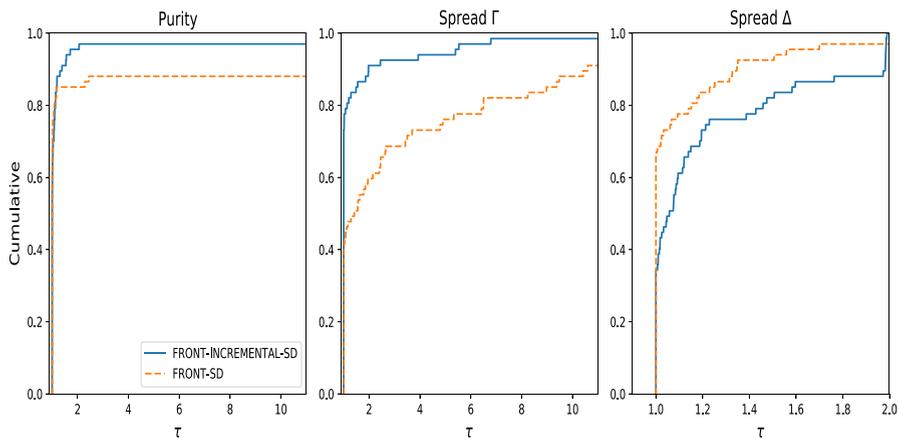


Fig. 2 Performance profiles of Purity and Spread metrics for FRONT-INCREMENTAL-SD and FRONT-SD for the 62 problems in which both reaches 20,000 function evaluations

As a second step, we compare the two variants of our algorithm, FRONT-INCREMENTAL-SD and FRONT-SD, in Fig. 1 from which the ability of FRONT-INCREMENTAL-SD to produce high quality solutions is clearly apparent.

On the contrary, since FRONT-SD is not able to move away from Pareto-stationary points, it is more efficient in terms of number of function evaluations, meaning that, in general, it does not reach the related stopping criterion.

Hence, for a more fair comparison in terms of function evaluations, we restrict the test set to the problems in which all the algorithms reached the maximum number of function evaluations (see Fig. 2). Although FRONT-INCREMENTAL-SD

slightly decreases its performance in terms of Purity, the global behaviour obtained on the whole dataset is confirmed.

Moreover, a further comparison between the two proposed variants has been reported. Figure 3 compares the performance of the two variants only on problems with 3 objective functions. The figure shows that the gap in terms of both Purity and Spread metrics becomes more relevant. This is an expected behaviour, since FRONT-INCREMENTAL-SD uses a set of search directions whose size exponentially increases with the number of objectives.

A further step of our numerical experimentation consists of separately comparing the best proposed variant, namely FRONT-INCREMENTAL-SD, with the MULTISTART-SD algorithm.

Since MULTISTART-SD algorithm is non deterministic, 10 independent runs for each problem were executed. For each problem, we merge the 10 retrieved Pareto fronts to a unique reference Pareto front by filtering out dominated solutions. Then, we identify the best run of the algorithm, namely MULTISTART-SD-BEST, as the one which retrieves the front with the best Purity score with respect to the reference Pareto front. Analogously, MULTISTART-SD-WORST can be defined in a similar way.

From Figs. 4 and 5 it emerges that FRONT-INCREMENTAL-SD is better than both versions of MULTISTART-SD algorithm with respect to all of the three metrics. This comes as no surprise since our algorithm employs a higher number of directions for searching new non-dominated solutions.

Finally, we have compared the performance of the best version FRONT-INCREMENTAL-SD with the algorithm NSGA-II [6] (using a Python wrapper implementation from PAGMO 2.1 package [2]). For the Pareto front approximation, we set up NSGA-II with a population of 100 points with 200 generations corresponding to a total of 20,000 objective function evaluations.

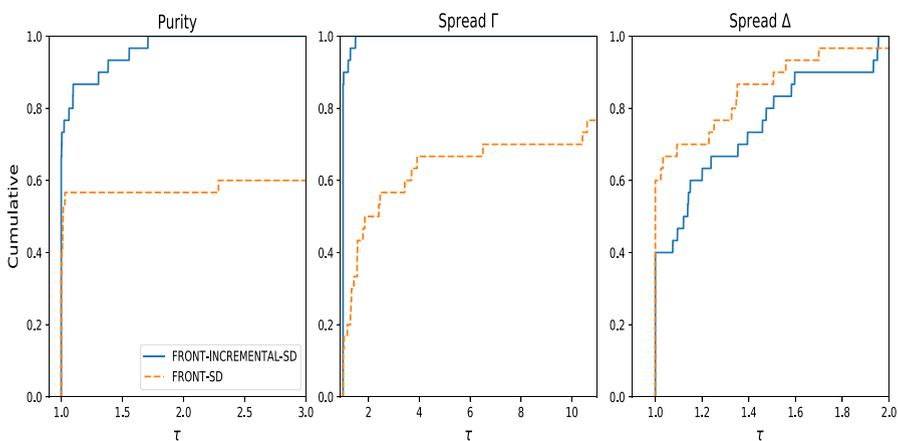


Fig. 3 Performance profiles of Purity and Spread metrics for FRONT-INCREMENTAL-SD and FRONT-SD only for problems with 3 objective functions

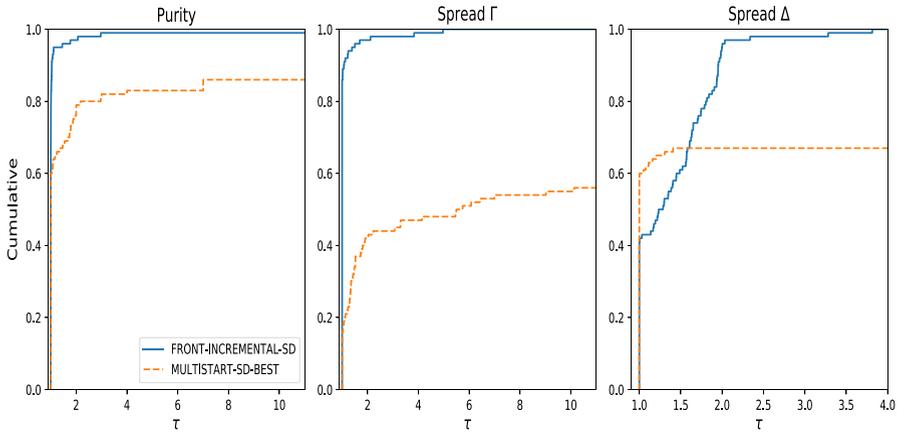


Fig. 4 Performance profiles of Purity and Spread metrics for FRONT-INCREMENTAL-SD and MULTISTART-SD-BEST

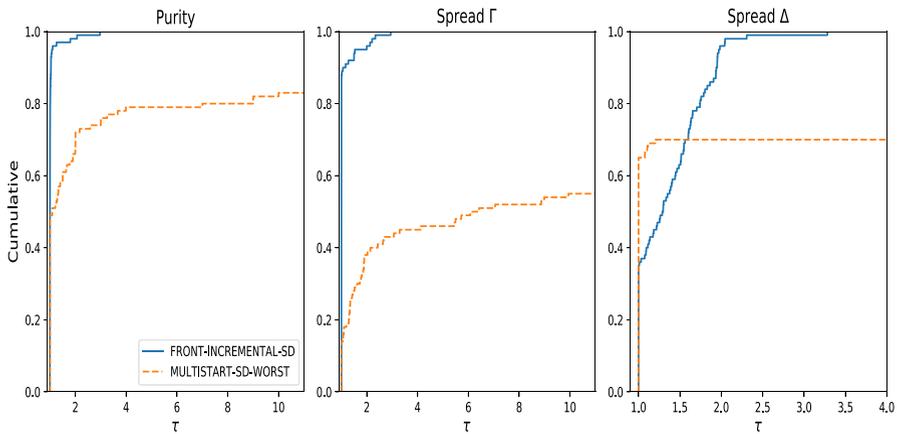


Fig. 5 Performance profiles of Purity and Spread metrics for FRONT-INCREMENTAL-SD and MULTISTART-SD-WORST

As for MULTISTART-SD algorithm, since also NSGA-II is non deterministic, we define NSGA-II-BEST and NSGA-II-WORST in the same way.

Since the performance profiles plots do not change significantly between the two cases of NSGA-II, only a comparison with NSGA-II-BEST has been reported. Figure 6, shows the performance profiles comparing the two solvers. Particularly, with respect to the Purity and Spread Γ metric, FRONT-INCREMENTAL-SD outperforms NSGA-II. In fact, since NSGA-II can generate at most 100 non-dominated solutions with this setup, the incremental algorithm extrapolates more directions with a better spanning of the space, resulting in better Purity and Spread Γ scores.

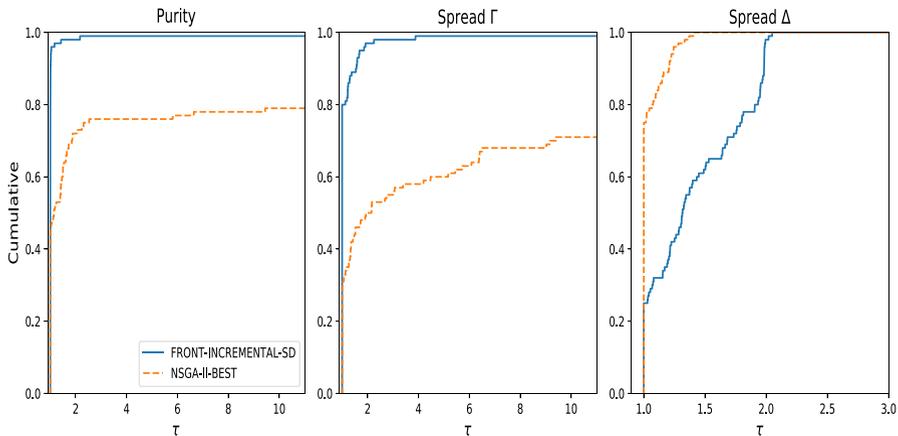


Fig. 6 Performance profiles of Purity and Spread metrics for FRONT-INCREMENTAL-SD and NSGA-II

On the contrary, NSGA-II outperforms FRONT-INCREMENTAL-SD with respect to the Spread Δ metric.

7 Concluding remarks

We present a steepest descent-based framework for smooth, unconstrained multiobjective optimization. The proposed framework is designed to generate an approximation of the Pareto front. To this aim, the algorithm generates sequences of sets of non-dominated points (instead of sequences of points) using steepest descent directions (not necessarily with respect to all the objective functions) and suitable extensions of Armijo-type linesearches. Global convergence results are stated and are based on a novel theoretical analysis that permits to characterize the properties of sequences of sets of points. The preliminary numerical results clearly show the validity and the effectiveness of the proposed approach compared with standard steepest descent methods.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., et al.: Tensorflow: large-scale machine learning on heterogeneous distributed systems. arXiv preprint [arXiv:1603.04467](https://arxiv.org/abs/1603.04467) (2016)
2. Biscani, F., Izzo, D., Hong, Y.C.: A global optimisation toolbox for massively parallel engineering optimisation. arXiv preprint [arXiv:1004.3824](https://arxiv.org/abs/1004.3824) (2010)
3. Cocchi, G., Liuzzi, G., Papini, A., Sciandrone, M.: An implicit filtering algorithm for derivative-free multiobjective optimization with box constraints. *Comput. Optim. Appl.* **69**(2), 267–296 (2018)
4. Custódio, A.L., Madeira, J.F.A., Vaz, A.I.F., Vicente, L.N.: Direct multisearch for multiobjective optimization. *SIAM J. Optim.* **21**(3), 1109–1140 (2011)

5. Deb, K., Mohan, M., Mishra, S.: Towards a Quick Computation of Well-Spread Pareto-Optimal Solutions. Springer, Berlin (2003)
6. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002)
7. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. *Math. Program.* **91**(2), 201–213 (2002)
8. Drummond, L.M.G., Maculan, N., Svaiter, B.F.: On the choice of parameters for the weighting method in vector optimization. *Math. Program.* **111**(1), 201–216 (2008)
9. Drummond, L.M.G., Svaiter, B.F.: A steepest descent method for vector optimization. *J. Comput. Appl. Math.* **175**(2), 395–414 (2005)
10. Eichfelder, G.: An adaptive scalarization method in multiobjective optimization. *SIAM J. Optim.* **19**(4), 1694–1718 (2009)
11. Fliege, J., Drummond, L.M.G., Svaiter, B.F.: Newton's method for multiobjective optimization. *SIAM J. Optim.* **20**(2), 602–626 (2009)
12. Fliege, J., Svaiter, B.F.: Steepest descent methods for multicriteria optimization. *Math. Methods Oper. Res.* **51**(3), 479–494 (2000)
13. Fliege, J., Vaz, A.I.F.: A method for constrained multiobjective optimization based on SQP techniques. *SIAM J. Optim.* **26**(4), 2091–2119 (2016)
14. Gen, M., Cheng, R., Lin, L.: *Multiobjective Genetic Algorithms*, pp. 1–47. Springer, Berlin (2008)
15. Jahn, J.: Scalarization in vector optimization. *Math. Program.* **29**(2), 203–218 (1984)
16. LLC Gurobi Optimization. *Gurobi Optimizer Reference Manual* (2018)
17. Liuzzi, G., Lucidi, S., Rinaldi, F.: A derivative-free approach to constrained multiobjective nonsmooth optimization. *SIAM J. Optim.* **26**(4), 2744–2774 (2016)
18. Miettinen, K.: *Nonlinear Multiobjective Optimization*. International Series in Operations Research & Management Science. Springer, Berlin (1998)
19. Povalej, I.: Quasi-Newton's method for multiobjective optimization. *J. Comput. Appl. Math.* **255**, 765–777 (2014)
20. Zhang, Q., Zhou, A., Zhao, S., Suganthan, P.N., Liu, W., Tiwari, S.: *Multiobjective Optimization Test Instances for the CEC 2009 Special Session and Competition*. (2008). <https://esa.github.io/pagmo2/docs/cpp/problems/cec2009.html>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.