



Cardinality-Constrained Multi-objective Optimization: Novel Optimality Conditions and Algorithms

Matteo Lapucci¹ · Pierluigi Mansueto¹ 

Received: 4 April 2023 / Accepted: 27 January 2024
© The Author(s) 2024

Abstract

In this paper, we consider multi-objective optimization problems with a sparsity constraint on the vector of variables. For this class of problems, inspired by the homonymous necessary optimality condition for sparse single-objective optimization, we define the concept of L -stationarity and we analyze its relationships with other existing conditions and Pareto optimality concepts. We then propose two novel algorithmic approaches: the first one is an iterative hard thresholding method aiming to find a single L -stationary solution, while the second one is a two-stage algorithm designed to construct an approximation of the whole Pareto front. Both methods are characterized by theoretical properties of convergence to points satisfying necessary conditions for Pareto optimality. Moreover, we report numerical results establishing the practical effectiveness of the proposed methodologies.

Keywords Multi-objective optimization · Cardinality constraints · Optimality conditions · Numerical methods · Pareto front

Mathematics Subject Classification 90C26 · 90C29 · 90C46

1 Introduction

Multi-objective optimization (MOO) is a mathematical tool that has received loads of attention by the research community for the last 25 years. As a matter of fact, MOO

Communicated by Ana Luisa Custodio.

✉ Pierluigi Mansueto
pierluigi.mansueto@unifi.it

Matteo Lapucci
matteo.lapucci@unifi.it

¹ University of Florence, Via di Santa Marta, 3, 50139 Florence, Italy

problems turned out to be relevant in different application fields, where objectives that are in contrast to each other must be taken into account (see, e.g., [11, 13, 25, 44, 46]).

With respect to single-objective optimization (SOO), the multi-objective case presents an additional complexity: in general, a solution minimizing all the objectives at once does not exist. This fact requires to introduce optimality conditions based on Pareto's theory, together with new, nontrivial, optimization schemes. Among the latter, scalarization [24, 43] and evolutionary [20, 36] approaches are very popular. Although these two algorithm classes have some appealing features, they also present important shortcomings. In particular, the scalarization approaches strongly depend on the problem domain and structure for the weight choices; moreover, even under strong regularity assumptions, they may lead to unbounded scalar problems for unfortunate weight selections [26, section 7]. On the other hand, despite excelling at handling very difficult tasks, convergence properties are hard to prove for evolutionary approaches [26].

A different branch of MOO algorithms that is receiving increasing attention is that of the descent algorithms [15, 22, 26, 27, 34]. These methods basically extend the classic SOO descent methods. For most of them, theoretical convergence properties have been proved. The earliest developed algorithms of this class were only able to produce a single Pareto stationary solution; in order to generate an approximation of the entire Pareto front, they were run multiple times in a multi-start fashion. More recently, some of these algorithms have been extended to overcome this limitation. These new approaches (see, e.g., [16–19, 33]) are capable of dealing with sequences of sets of points and thus producing a Pareto front approximation in an efficient and effective manner. Inspired by works for SOO [38, 41], front-oriented descent methods have also been combined with genetic algorithms for MOO (see, e.g., [35]).

A second topic recently investigated by the optimization community concerns problems where solutions with few nonzero components are required [47]. Solution sparsity is often induced by the direct introduction of a cardinality constraint on the variables vector. However, setting an upper bound for the ℓ_0 pseudo-norm makes the problem partly combinatorial and thus \mathcal{NP} -hard [8, 42]. For this reason, many approaches whose aim is to solve this problem approximately have been proposed. We refer the reader to [47] for a thorough survey of these methods. However, algorithms dealing exactly with the ℓ_0 pseudo-norm can be found in the literature. In particular, the *iterative hard thresholding* (IHT) algorithm [1], the *penalty decomposition* (PD) approach [39] and the *sparse neighborhood search* (SNS) method [32] are designed to be employable in the most general cases, even without convexity assumptions. With these methods, problems are tackled by means of continuous local optimization steps and convergence to solutions satisfying necessary optimality conditions is guaranteed.

Although the two challenges have been thoroughly investigated separately, the combination of sparsity and multiple objectives has almost not been explored. The theoretical foundation for cardinality-constrained MOO was recently laid in [31], where a penalty decomposition approach was also proposed for sparse MOO tasks along with its convergence analysis. Moreover, a theoretical study extending the work from [10] to the MOO case was presented in [28]. The development of high-performing procedures to deal with this class of problems is beneficial for many real-world applications. For instance, there are several reasons in machine learning for inducing sparsity within

classification/regression models (e.g., interpretability [4], robustness [48], lightness [12]). In addition, there are approaches in the literature where learning tasks can be tackled from a Pareto-based, multi-objective perspective: Fitting quality and model complexity are just two examples of conflicting objectives for which a good trade-off may be useful [29].

In this paper, we continue the theoretical analysis started in [31], introducing new optimality conditions for MOO problems with cardinality constraints. In particular, we define the concept of L -stationarity in MOO, which is directly inspired by the homonymous condition for sparse SOO tasks [1]. Then, we introduce two new algorithms to solve these problems: The first one consists of an extension of the IHT method to the MOO case and it is designed to retrieve an L -stationary solution; we call this method *multi-objective iterative hard thresholding* (MOIHT) and prove that it is indeed guaranteed to converge to points satisfying the newly introduced necessary optimality condition; and the second algorithm, on the other hand, is a two-stage approach whose ultimate goal is to approximate the whole Pareto front. This method, which we call *sparse front steepest descent* (SFSD), is also analyzed theoretically and then shown, numerically, to actually reconstruct well the solution sets.

The remainder of the paper is organized as follows. In Sect. 2, we first review the main MOO concepts along with some notions for cardinality-constrained MOO problems. In Sect. 3, we define L -stationarity for the considered class of problems and state its theoretical relations with Pareto's theory and other existing optimality conditions. In Sect. 4, we provide a description of the MOIHT algorithm, along with its convergence analysis; moreover, in this section we propose the SFSD methodology. In Sect. 5, we report the results of some computational experiments, showing the goodness of our novel approaches. Finally, in Sect. 6, we provide some concluding remarks.

2 Preliminaries

In this paper, we consider problems of the following form:

$$\min_{x \in \mathbb{R}^n} F(x) = (f_1(x), \dots, f_m(x))^T \quad \text{s.t.} \quad \|x\|_0 \leq s, \quad (1)$$

where $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is a continuously differentiable vector-valued function, $\|\cdot\|_0$ denotes the ℓ_0 pseudo-norm, i.e., the number of nonzero components of a vector, and $s \in \mathbb{N}$ is such that $1 \leq s < n$. In what follows, we indicate with $\|\cdot\|$ the Euclidean norm in \mathbb{R}^n . We denote by $\Omega = \{x \in \mathbb{R}^n \mid \|x\|_0 \leq s\}$ the closed and nonempty set induced by the upper bound on the ℓ_0 pseudo-norm.

To deal with the multi-objective setting, we need to define a partial ordering in \mathbb{R}^m : given $u, v \in \mathbb{R}^m$, we say that $u < v$ if and only if, for all $j \in \{1, \dots, m\}$, $u_j < v_j$; an analogous definition can be stated for the operators $\leq, >, \geq$. Furthermore, given the objective function F , we say that x *dominates* y w.r.t. F if $F(x) \leq F(y)$ and $F(x) \neq F(y)$ and we denote it by $F(x) \preceq F(y)$.

In multi-objective optimization, a solution which simultaneously minimizes all the objectives is unlikely to exist. In this case, optimality concepts are based on Pareto’s theory.

Definition 2.1 A point $\bar{x} \in \Omega$ is *Pareto optimal* for problem (1) if there does not exist any $y \in \Omega$ such that $F(y) \preceq F(\bar{x})$. If there exists a neighborhood $\mathcal{N}(\bar{x})$ such that the property holds in $\Omega \cap \mathcal{N}(\bar{x})$, then \bar{x} is *locally Pareto optimal*.

Pareto optimality is a strong property, and as a consequence, it is often hard to achieve in practice. Then, a weaker but more affordable condition can be introduced.

Definition 2.2 A point $\bar{x} \in \Omega$ is *weakly Pareto optimal* for problem (1) if there does not exist any $y \in \Omega$ such that $F(y) < F(\bar{x})$. If there exists a neighborhood $\mathcal{N}(\bar{x})$ such that the property holds in $\Omega \cap \mathcal{N}(\bar{x})$, then \bar{x} is *locally weakly Pareto optimal*.

We refer to the set of Pareto optimal solutions as the *Pareto set*; the image of the latter under F is called *Pareto front*.

Additional notation: Given an index set $S \subseteq \{1, \dots, n\}$, the cardinality of S is indicated with $|S|$, while we denote by $\bar{S} = \{1, \dots, n\} \setminus S$ its *complementary set*; we call S a *singleton* if $|S| = 1$. Letting $x \in \mathbb{R}^n$, we denote by x_S the sub-vector of x induced by S , i.e., the vector composed by the components x_i , with $i \in S$; $S_1(x) = \{i \in \{1, \dots, n\} \mid x_i \neq 0\}$ represents the *support set* of x , that is, the set of the indices corresponding to the nonzero components of x ; $S_0(x) = \{1, \dots, n\} \setminus S_1(x)$ is the $S_1(x)$ complementary set. Furthermore, according to [2], we say that an index set J is a *super support set* for x if $S_1(x) \subseteq J$ and $|J| = s$; the set of all super support sets at x is denoted by $\mathcal{J}(x)$ and it is a singleton if and only if $\|x\|_0 = s$. Finally, we denote by $\mathbf{1}_N$ and $\mathbf{0}_N$, with $N \in \mathbb{N} \setminus \{0\}$, the N -dimensional vectors of all ones and all zeros, respectively.

2.1 The Proximal Operator in Multi-objective Optimization

A thorough analysis of proximal methods in the multi-objective setting can be found in the literature (see, e.g., [9, 45]). For the scope of this work, we refer to the discussion carried out in [45], where the considered MOO problems are of the form

$$\min_{x \in \mathbb{R}^n} (f_1(x) + g_1(x), \dots, f_m(x) + g_m(x))^T. \tag{2}$$

For all $j \in \{1, \dots, m\}$, f_j is assumed to be continuously differentiable, whereas g_j is lower semi-continuous, proper convex but not necessarily smooth.

Let $x_k \in \mathbb{R}^n$. A *proximal step* at x_k can be carried out according to $x_{k+1} = x_k + t_k d_k$, where t_k is a suitable step size and the descent direction d_k is obtained solving

$$\min_{d \in \mathbb{R}^n} \max_{j \in \{1, \dots, m\}} \left\{ \nabla f_j(x_k)^T d + g_j(x_k + d) - g_j(x_k) \right\} + \frac{L}{2} \|d\|^2, \tag{3}$$

where $L > 0$. An optimal solution of problem (2) is such that $\mathbf{0}_n$ is solution to (3).

Interestingly and similarly to the scalar case, problem (3) can be seen as a generalization of well-known schemes to define the search direction:

- if, for all $j \in \{1, \dots, m\}$, $g_j = 0$, then (3) coincides with the problem of finding the steepest common descent direction [27];
- if, for all $j \in \{1, \dots, m\}$, g_j is the indicator function of a convex set C , then (3) becomes equivalent to the projected gradient direction problem [22].

In the next section, we are going to show that the proximal operator can be used to handle the nonconvex set Ω , in line with the work [1] for scalar optimization.

3 Optimality Conditions

Under differentiability assumptions on the objective function F , a Pareto stationarity condition was proved in [31] to be necessary for (local) weak Pareto optimality. In what follows, we report slightly different definition and properties, adapted to problem (1).

Definition 3.1 [31, Definition 3.2] A point $\bar{x} \in \Omega$ is Pareto stationary for (1) if

$$\theta(\bar{x}) = \min_{d \in \mathcal{D}(\bar{x})} \max_{j=1, \dots, m} \nabla f_j(\bar{x})^\top d + \frac{1}{2} \|d\|^2 = 0, \tag{4}$$

where $\mathcal{D}(\bar{x}) = \{v \in \mathbb{R}^n \mid \exists \bar{t} > 0 : \bar{x} + tv \in \Omega \ \forall t \in [0, \bar{t}]\} = \{v \in \mathbb{R}^n \mid \|v_{S_0(\bar{x})}\|_0 \leq s - \|\bar{x}\|_0\}$ is the set of feasible directions at \bar{x} .

We denote by $v(\bar{x})$ the set of optimal solutions of problem (4) at \bar{x} .

Lemma 3.1 [31, Proposition 3.3] *Let $\bar{x} \in \Omega$ be locally weakly Pareto optimal for problem (1). Then, \bar{x} is Pareto stationary for (1).*

The second lemma states that, assuming the convexity of the objective functions, the stationarity condition is also sufficient for local weak Pareto optimality.

Lemma 3.2 *Assume F is component-wise convex. Let $\bar{x} \in \Omega$ a Pareto stationary point for problem (1). Then, \bar{x} is locally weakly Pareto optimal for (1).*

Proof See Appendix A. □

Moreover, in [31], the Lu–Zhang first-order optimality conditions for scalar cardinality-constrained problems [39] have been extended to the multi-objective optimization setting.

Definition 3.2 [31, Definition 3.6] A point $\bar{x} \in \Omega$ satisfies the *multi-objective Lu–Zhang first-order optimality conditions* (MOLZ conditions) for (1) if there exists a super support set $J \in \mathcal{J}(\bar{x})$ such that

$$\theta_J(\bar{x}) = \min_{d \in \mathbb{R}^n} \max_{j=1, \dots, m} \nabla f_j(\bar{x})^\top d + \frac{1}{2} \|d\|^2 = 0 \quad \text{s.t.} \quad d_{\bar{J}} = \mathbf{0}_{|\bar{J}|} \tag{5}$$

Since problem (5) has a strongly convex objective function and a convex feasible set, it has a unique optimal solution at \bar{x} that we indicate with $d_J(\bar{x})$.

Lemma 3.3 [31, Proposition 3.7] *Let $\bar{x} \in \Omega$ be a Pareto stationary point for problem (1). Then, \bar{x} satisfies MOLZ conditions.*

As pointed out in [31], the converse is not always true; in order to obtain an equivalence between the two conditions, we need a stronger requirement.

Lemma 3.4 [31, Proposition 3.10] *A point $\bar{x} \in \Omega$ is a Pareto stationary point for problem (1) if and only if it satisfies MOLZ conditions for all $J \in \mathcal{J}(\bar{x})$.*

The Pareto stationarity condition can be interpreted as a direct extension of the *basic feasibility* concept in cardinality-constrained SOO [1, 2]. As such, the limitations of scalar basic feasibility naturally get transferred to the MOO case; in particular, Pareto stationarity is only a local optimality condition and it does not allow to obtain information about the quality of the current support set. The MOLZ conditions emphasize this issue even more, being generally less restrictive than Pareto stationarity.

With the above consideration in mind, we are motivated to extend the stronger L -stationarity condition from [1] to the MOO case. In order to do so, we shall reinterpret L -stationarity in terms of proximal operators. Specifically, we can employ problem (3) to define L -stationarity for MOO.

Let us consider the problem

$$\theta_L(\bar{x}) = \min_{d \in \mathcal{D}_L(\bar{x})} \max_{j=1, \dots, m} \nabla f_j(\bar{x})^\top d + \frac{L}{2} \|d\|^2, \tag{6}$$

where $\mathcal{D}_L(\bar{x}) = \{v \in \mathbb{R}^n \mid \bar{x} + v \in \Omega\}$ and let us denote by $v_L(\bar{x})$ the set of optimal solutions at \bar{x} (since Ω is not a convex set, the solution is not necessarily unique). It is easy to notice that problem (6) is equivalent to (3) where, for all $j \in \{1, \dots, m\}$, g_j is the indicator function of the set Ω .

Lemma 3.5 *Let $\bar{x} \in \Omega$. Then, the following conditions hold:*

- (i) $\theta_L(\bar{x})$ and $v_L(\bar{x})$ are well defined;
- (ii) $\theta_L(\bar{x}) \leq 0$;
- (iii) the mapping $x \rightarrow \theta_L(x)$ is continuous.

Proof (i) The proof is trivial since

- the feasible set $\mathcal{D}_L(\bar{x})$ is closed and nonempty,
- $\max_{j=1, \dots, m} \nabla f_j(\bar{x})^\top d + \frac{L}{2} \|d\|^2$ is strongly convex and continuous in $\mathcal{D}_L(\bar{x})$.

(ii) Given $\hat{d} = \mathbf{0}_n$, we have that $\max_{j=1, \dots, m} \nabla f_j(\bar{x})^\top \hat{d} + \frac{L}{2} \|\hat{d}\|^2 = 0$. Since $\hat{d} \in \mathcal{D}_L(\bar{x})$, we get the thesis.

(iii) The proof is identical to the one of Proposition 4 in [22]. The argument is not spoiled by the set $\mathcal{D}_L(\bar{x})$ being nonconvex. □

We are now ready to introduce the definition of L -stationarity in MOO.

Definition 3.3 A point $\bar{x} \in \Omega$ is L -stationary for problem (1) if $\theta_L(\bar{x}) = 0$.

Remark 3.1 By simple algebraic manipulations, the problem in (6) can be rewritten as

$$\min_{z \in \Omega} \max_{j=1, \dots, m} \nabla f_j(\bar{x})^\top (z - \bar{x}) + \frac{L}{2} \|z - \bar{x}\|^2. \tag{7}$$

We can now observe that, if $m = 1$, Definition 3.3 actually coincides with scalar L -stationarity. Indeed, exploiting (7), $\theta_L(\bar{x})$ is equivalent to

$$\min_{z \in \Omega} \nabla f(\bar{x})^\top (z - \bar{x}) + \frac{L}{2} \|z - \bar{x}\|^2 = \min_{z \in \Omega} \frac{L}{2} \|z - \bar{x}\|^2 + \frac{1}{L} \|\nabla f(\bar{x})\|^2 - \frac{1}{2L} \|\nabla f(\bar{x})\|^2.$$

The minimum in the above problem is attained for $z^* \in \Pi_\Omega[\bar{x} - \frac{1}{L} \nabla f(\bar{x})]$, with Π_Ω being the (not unique) Euclidean projection onto the nonconvex set Ω . We thus have that $\theta_L(\bar{x}) = 0$ if $\frac{L}{2} \|z^* - \bar{x}\|^2 + \nabla f(\bar{x})^\top (z^* - \bar{x}) = 0$, which is satisfied if $\bar{x} \in \Pi_\Omega[\bar{x} - \frac{1}{L} \nabla f(\bar{x})]$, i.e., \bar{x} is L -stationary according to [1].

In the rest of the section, we analyze the relations between L -stationarity, Pareto optimality, Pareto stationarity and MOLZ conditions. We begin showing that, for any $L > 0$, each L -stationary point is Pareto stationary.

Proposition 3.1 *Let $\bar{x} \in \Omega$ be an L -stationary point for problem (1) with $L > 0$. Then, \bar{x} is Pareto stationary for (1).*

Proof By contradiction, we assume that \bar{x} is not Pareto stationary for (1), i.e., there exists $\hat{d} \in \mathcal{D}(\bar{x})$ such that

$$0 > \max_{j \in \{1, \dots, m\}} \nabla f_j(\bar{x})^\top \hat{d} + \frac{1}{2} \|\hat{d}\|^2 \geq \max_{j \in \{1, \dots, m\}} \nabla f_j(\bar{x})^\top \hat{d}, \tag{8}$$

where the second inequality is justified by the nonnegativity of the norm operator.

We now define the direction $\tilde{d}(t) = t\hat{d}$. Given the definition of $\mathcal{D}(\bar{x})$ and the feasibility of \hat{d} , we have there exists $\bar{t} > 0$ such that $\bar{x} + \tilde{d}(t) \in \Omega \forall t \in [0, \bar{t}]$. Thus, by definition of $\mathcal{D}_L(\bar{x})$, for all $t \in [0, \bar{t}]$, $\tilde{d}(t) \in \mathcal{D}_L(\bar{x})$. Let us define the function $\tilde{\theta}_L : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ as $\tilde{\theta}_L(x, d) = \max_{j=1, \dots, m} \nabla f_j(x)^\top d + \frac{L}{2} \|d\|^2$. By (6), it follows that $\theta_L(\bar{x}) = \tilde{\theta}_L(\bar{x}, \tilde{d}^L)$, where $\tilde{d}^L \in v_L(\bar{x})$, and also

$$\theta_L(\bar{x}) \leq \tilde{\theta}_L(\bar{x}, d), \quad \forall d \in \mathcal{D}_L(\bar{x}). \tag{9}$$

Combining the definitions of $\tilde{d}(t)$ and $\tilde{\theta}_L(x, d)$, we get that $\tilde{\theta}_L(\bar{x}, \tilde{d}(t)) = t \max_{j=1, \dots, m} \nabla f_j(\bar{x})^\top \hat{d} + t^2 \frac{L}{2} \|\hat{d}\|^2$. It is easy to see that $\tilde{\theta}_L(\bar{x}, \tilde{d}(t)) < 0$ if

$$0 < t < -\frac{2}{L \|\hat{d}\|^2} \max_{j=1, \dots, m} \nabla f_j(\bar{x})^\top \hat{d}, \tag{10}$$

where the right-hand side is a positive quantity as $L > 0$ and (8) holds.

Then, taking into account the feasibility of $\tilde{d}(t)$, (9) and (10), we can define a direction $\tilde{d}(\hat{t})$, with $\hat{t} \in \left(0, \min\{\bar{t}, -\frac{2}{L\|\hat{d}\|^2} \max_{j=1,\dots,m} \nabla f_j(\bar{x})^\top \hat{d}\}\right)$, so that $\tilde{d}(\hat{t}) \in \mathcal{D}_L(\bar{x})$ and $\theta_L(\bar{x}) \leq \tilde{\theta}_L(\bar{x}, \tilde{d}(\hat{t})) < 0$. We finally get a contradiction since, by hypothesis, \bar{x} is an L -stationary point for (1), i.e., $\theta_L(\bar{x}) = 0$. \square

The last result also highlights the relation between L -stationarity and MOLZ conditions, which, as stated in Lemma 3.3, are necessary for Pareto stationarity. We formalize it in the next corollary.

Corollary 3.1 *Let $\bar{x} \in \Omega$ be an L -stationary point for problem (1) with $L > 0$. Then, \bar{x} satisfies the MOLZ conditions.*

Given Proposition 3.1 and Lemma 3.2, we can also state that, under convexity assumptions for F , L -stationarity is a sufficient condition for local weak Pareto optimality.

Corollary 3.2 *Assume that F is component-wise convex and let $\bar{x} \in \Omega$ be an L -stationary point for problem (1) with $L > 0$. Then, \bar{x} is locally weakly Pareto optimal for (1).*

In order to continue the analysis, we need to introduce a couple of notions. The first one is an assumption similar to the one used for L -stationarity in [1], while the second one concerns an adaptation of the *descent lemma* to MOO.

Assumption 1 For all $j \in \{1, \dots, m\}$, ∇f_j is Lipschitz-continuous over \mathbb{R}^n with constant $L(f_j)$, i.e., $\|\nabla f_j(x) - \nabla f_j(y)\| \leq L(f_j)\|x - y\|$ for all $x, y \in \mathbb{R}^n$.

In what follows, we indicate with $L(F) \in \mathbb{R}^m$ the vector of the Lipschitz constants, i.e., $L(F) = (L(f_1), \dots, L(f_m))^\top$.

Lemma 3.6 [3, Proposition A.24] *Let $f_j, j = 1, \dots, m$, be a continuously differentiable function satisfying Assumption 1. Then, for all $L \geq L(f_j)$ and any $x, d \in \mathbb{R}^n$, we have that $f_j(x + d) \leq f_j(x) + \nabla f_j(x)^\top d + \frac{L}{2}\|d\|^2$.*

We are ready to show that, for specific L values, the L -stationarity condition is necessary for weak Pareto optimality.

Proposition 3.2 *Let Assumption 1 hold, $\bar{x} \in \Omega$ be a weakly Pareto optimal point for problem (1) and $L > \max_{j=1,\dots,m} L(f_j)$. Then, \bar{x} is L -stationary for (1). Moreover, we have that $v_L(\bar{x}) = \{\mathbf{0}_n\}$, i.e., the set $v_L(\bar{x})$ is a singleton.*

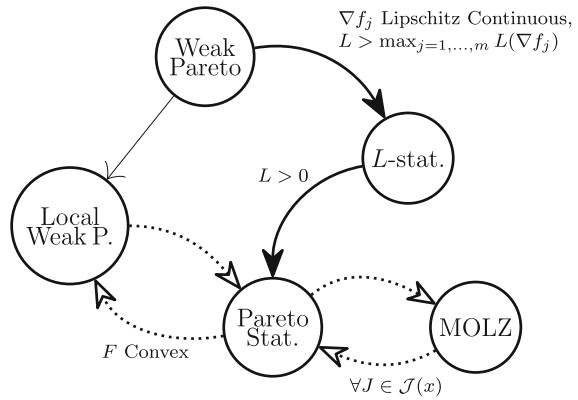
Proof By contradiction, let us assume that either \bar{x} is not L -stationary for (1) or $v_L(\bar{x}) \setminus \{\mathbf{0}_n\} \neq \emptyset$. Then, there exists a direction $\hat{d} \in \mathcal{D}_L(\bar{x})$ such that $\hat{d} \neq \mathbf{0}_n$ and

$$\max_{j \in \{1,\dots,m\}} \nabla f_j(\bar{x})^\top \hat{d} + \frac{L}{2}\|\hat{d}\|^2 \leq 0. \tag{11}$$

By Lemma 3.6, we have that, for all $h \in \{1, \dots, m\}$,

$$f_h(\bar{x} + \hat{d}) \leq f_h(\bar{x}) + \nabla f_h(\bar{x})^\top \hat{d} + \frac{L(f_h)}{2}\|\hat{d}\|^2. \tag{12}$$

Fig. 1 Graphical scheme of the theoretical relationships among (local) weak Pareto optimality, L -stationarity, Pareto stationarity and MOLZ conditions. The L -stationarity properties are displayed with solid arrows, while the other ones, stated and proved in [31], are indicated with dotted arrows



From Equation (11), we get that $\nabla f_h(\bar{x})^\top \hat{d} \leq \max_{j \in \{1, \dots, m\}} \nabla f_j(\bar{x})^\top \hat{d} \leq -\frac{L}{2} \|\hat{d}\|^2$, where the first inequality comes from the definition of maximum operator. Recalling the hypothesis on L and the nonnegativity of the norm, we combine (12) and the last result obtaining that

$$f_h(\bar{x} + \hat{d}) \leq f_h(\bar{x}) + \frac{L(f_h) - L}{2} \|\hat{d}\|^2 < f_h(\bar{x}) + \frac{L(f_h) - \max_{j \in \{1, \dots, m\}} L(f_j)}{2} \|\hat{d}\|^2.$$

Thus, for all $h \in \{1, \dots, m\}$ we have $f_h(\bar{x} + \hat{d}) - f_h(\bar{x}) < \frac{\|\hat{d}\|^2}{2} (L(f_h) - \max_{j \in \{1, \dots, m\}} L(f_j)) \leq 0$, leading to the conclusion that we have found a point $\bar{x} + \hat{d} \in \Omega$ such that $F(\bar{x} + \hat{d}) < F(\bar{x})$. This is a contradiction since, by hypothesis, \bar{x} is weakly Pareto optimal for (1). Thus, we get the thesis. \square

All the theoretical relationships stated in this section are shown in a graphical and compact view in Fig. 1. The analysis on L -stationarity highlights how the choice of the L value could be crucial: if L is too small, L -stationarity might not be a necessary optimality condition; on the other hand, if L gets too large, all the Pareto stationary points also become L -stationary. This behavior can be better noticed with an example.

Example 3.1 Let us consider the following optimization problem:

$$\min_{x \in \mathbb{R}^2} \frac{1}{2} \left((x_1 - 3)^2 + (x_2 - 2.5)^2, (x_1 - 1)^2 + (x_2 - 0.5)^2 \right)^\top \text{ s.t. } \|x\|_0 \leq 1.$$

The Lipschitz constant of the gradient of both objective functions f_j is $L(f_j) = 1$. In Fig. 2, the Pareto optimal solutions and the Pareto front are plotted: the problem has global optimal solutions corresponding to points with $x_1 \neq 0$; the local ones are characterized by the second component $x_2 \neq 0$. By Lemmas 3.1–3.3, it follows that all the considered points are Pareto stationary and satisfy the MOLZ conditions. In Fig. 3, we show which Pareto solutions are L -stationary, considering four different choices for L . If L is chosen too small (Fig. 3a), some global Pareto solutions do not result to

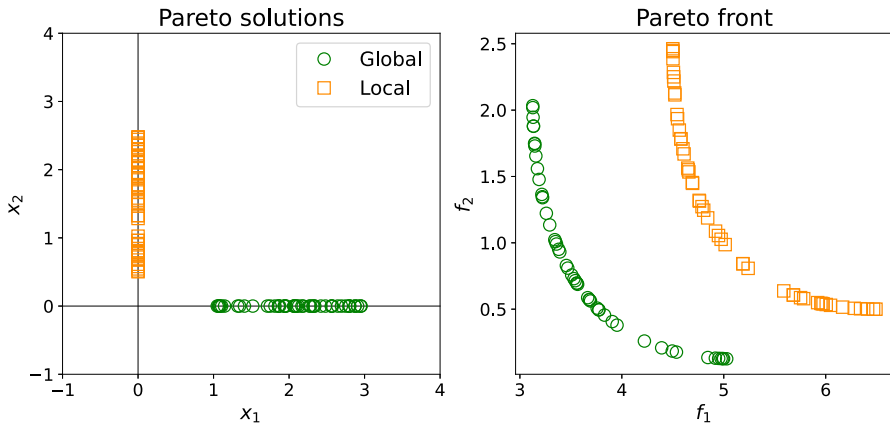


Fig. 2 Pareto optimal solutions and Pareto front of problem of Example 3.1

be L -stationary. As stated in Proposition 3.2, the L -stationarity condition turns out to be necessary for Pareto optimality for an L value greater than the Lipschitz constants (Fig. 3b where $L = 1.01$). On the other hand, a too high value makes the condition rather weak: in Fig. 3c ($L = 1.25$), even some local Pareto solutions are L -stationary. The situation is further stressed in Fig. 3d where $L = 2$ and all Pareto stationary points are also L -stationary.

4 New Algorithmic Approaches for Sparse MOO Problems

In this section, we propose two procedures to solve cardinality-constrained MOO problems. The first one can be seen as an extension of the *iterative hard thresholding* (IHT) algorithm [1]; the second one is a front approximation approach that takes as input candidate solutions, possibly associated with different support sets, and then spans the portions of the Pareto front associated with those supports. We report their schemes and discuss their properties in separate sections.

4.1 Multi-objective Iterative Hard Thresholding

The first procedure we introduce is the *multi-objective iterative hard thresholding* algorithm. In the remainder of the paper, we refer to it as MOIHT. The scheme of the method is reported in Algorithm 1. At each iteration of MOIHT, the current solution x_k is updated solving problem (13). The execution continues until an L -stationary point for (1) is found.

Remark 4.1 At each iteration k , the solution x_{k+1} generated by MOIHT is feasible for (1). Indeed, the feasibility easily follows by definition of $\mathcal{D}_L(x_k)$.

Remark 4.2 It is very important to underline that Step 4 is a practical operation that can be effectively implemented in the general case. Problem (13) can indeed be solved up

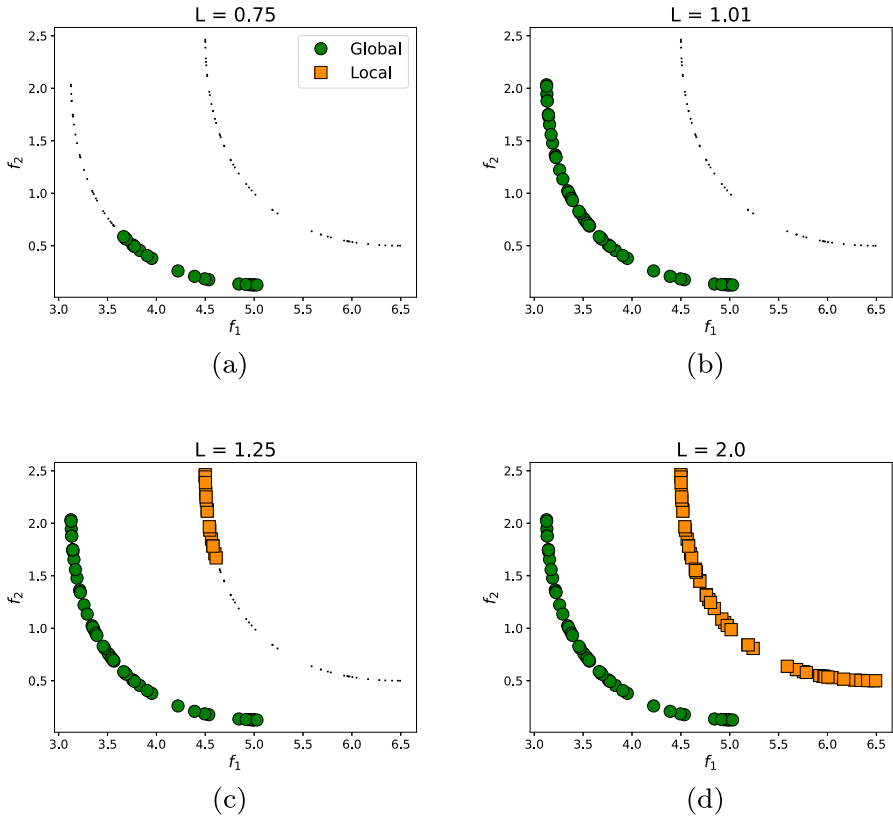


Fig. 3 L -stationary points in the Pareto front of problem of Example 3.1 ($L(F) = [1, 1]^T$) for different values of L

to global optimality, for example with mixed-integer programming techniques (see, e.g., [6, 7]). Defining a sufficiently large scalar $M > 0$, the problem can be equivalently reformulated as

$$\begin{aligned} \min_{t, d, \delta} \quad & t + \frac{L}{2} \|d\|^2 \quad \text{s.t.} \quad \nabla f_j(\bar{x})^\top d \leq t \quad \forall j = 1, \dots, m, \quad \mathbf{1}_n^\top \delta \leq s, \\ & -M\delta \leq \bar{x} + d \leq M\delta, \quad t \in \mathbb{R}, d \in \mathbb{R}^n, \delta \in \{0, 1\}^n. \end{aligned}$$

4.1.1 Convergence Analysis

In this section, we analyze the method from a theoretical perspective. Before proving the main convergence result, we need to state an additional assumption on F and prove a technical lemma.

Algorithm 1: Multi-Objective Iterative Hard Thresholding

```

1 Input:  $x_0 \in \Omega$ ,  $L > \max_{j \in \{1, \dots, m\}} L(f_j)$ .
2  $k = 0$ 
3 while  $x_k$  is not  $L$ -stationary for problem (1) do
4   Compute
      
$$v_L(x_k) = \arg \min_{d \in \mathcal{D}_L(x_k)} \max_{j \in \{1, \dots, m\}} \nabla f_j(x_k)^\top d + \frac{L}{2} \|d\|^2 \quad (13)$$

5   Let  $d_k^L \in v_L(x_k)$ 
6    $x_{k+1} = x_k + d_k^L$ 
7   Let  $k = k + 1$ 
8 return  $x_k$ 

```

Assumption 2 F has bounded level sets in the multi-objective sense, i.e., the set $\mathcal{L}_F(z) = \{x \in \Omega \mid F(x) \leq z\}$ is bounded for any $z \in \mathbb{R}^m$.

Lemma 4.1 Let Assumptions 1–2 hold and $\{x_k\}$ be the sequence generated by Algorithm 1 with constant $L > \max_{j \in \{1, \dots, m\}} L(f_j)$. Then:

- (i) for all k , $F(x_k) - F(x_{k+1}) \geq \frac{1}{2} \|x_k - x_{k+1}\|^2 (\mathbf{1}_m L - L(F))$;
- (ii) for all k , if $x_k \neq x_{k+1}$, then $F(x_{k+1}) < F(x_k)$;
- (iii) for all $j \in \{1, \dots, m\}$, the sequence $\{f_j(x_k)\}$ is nonincreasing;
- (iv) the sequence $\{F(x_k)\}$ converges;
- (v) $\lim_{k \rightarrow \infty} \|x_k - x_{k+1}\|^2 = 0$.

Proof (i) The thesis can be proved making an argument similar to that of Proposition 3.2 and reminding that $x_{k+1} = x_k + d_k^L$, with $d_k^L \in v_L(x_k)$ (Step 6 of Algorithm 1).

- (ii) It follows directly from Point (i), recalling that $L > L(f_j)$ for all $j \in \{1, \dots, m\}$ and $\|x_k - x_{k+1}\| > 0$.
- (iii) By Point (i) and the hypothesis on L , we have that, for all k and $j \in \{1, \dots, m\}$, $f_j(x_{k+1}) \leq f_j(x_k)$. Thus, for all $j \in \{1, \dots, m\}$, the sequence $\{f_j(x_k)\}$ is nonincreasing.
- (iv) It follows from Assumption 2 and Point (iii).
- (v) From Point (i), we have that, for all k and $j \in \{1, \dots, m\}$, $\frac{L-L(f_j)}{2} \|x_k - x_{k+1}\|^2 \leq f_j(x_k) - f_j(x_{k+1})$. Since f_j is continuous, we can take the limit for $k \rightarrow \infty$ on both sides of the inequality: $\lim_{k \rightarrow \infty} \frac{L-L(f_j)}{2} \|x_k - x_{k+1}\|^2 \leq \lim_{k \rightarrow \infty} f_j(x_k) - f_j(x_{k+1}) = 0$, where the equality comes from Point (iv). By the definition of L and the nonnegativity of the norm, the statement is proved. \square

Proposition 4.1 Let Assumptions 1–2 hold and $\{x_k\}$ be the sequence generated by Algorithm 1 with constant $L > \max_{j \in \{1, \dots, m\}} L(f_j)$. Then, the sequence admits cluster points, each one being L -stationary for problem (1).

Proof First, we prove that the sequence admits limit points. By Lemma 4.1, we deduce that, for all k , $F(x_k) \leq F(x_{k-1}) \leq \dots \leq F(x_0)$. Moreover, as noted in Remark 4.1, $x_k \in \Omega$ for all k . Thus, we have that $x_k \in \mathcal{L}_F(F(x_0)) \forall k$. Since Assumption 2 holds, we conclude that the sequence $\{x_k\}$ is bounded and it thus admits limit points.

Now, we denote as \bar{x} a limit point, i.e., there exists a subsequence $K \subseteq \{0, 1, \dots\}$ such that

$$\lim_{\substack{k \rightarrow \infty \\ k \in K}} x_k = \bar{x}. \quad (14)$$

By Point (v) of Lemma 4.1 and Step 6, we have that $\lim_{k \rightarrow \infty, k \in K} \|d_k^L\|^2 = 0$, and thus, $d_k^L \rightarrow_K \mathbf{0}_n$. Considering this last result and taking the limit for $k \rightarrow \infty, k \in K$ in problem (13), by Point (iii) of Lemma 3.5, Steps 4–5, the continuous differentiability of F and the continuity of the maximum and norm operators, we get that

$$\theta_L(\bar{x}) = \lim_{\substack{k \rightarrow \infty \\ k \in K}} \theta_L(x_k) = \lim_{\substack{k \rightarrow \infty \\ k \in K}} \max_{j \in \{1, \dots, m\}} \nabla f_j(x_k)^\top d_k^L + \frac{L}{2} \|d_k^L\|^2 = 0.$$

We conclude that $\theta_L(\bar{x}) = 0$, and then, \bar{x} is L -stationary for problem (1). \square

Remark 4.3 By Proposition 4.1 and the continuity of θ_L (Lemma 3.5) we are guaranteed that, for any $\varepsilon > 0$, Algorithm 1 will produce a point x_k such that $\theta_L(x_k) > -\varepsilon$ in a finite number of iterations. Thus, we can effectively employ this condition as a practical stopping criterion for the MOIHT procedure.

4.2 Sparse Front Steepest Descent

In what follows, we describe and analyze the *sparse front steepest descent* (SFSD) methodology. The algorithm can be seen as a two phases approach, which is based on the following consideration: in problems of the form (1), the Pareto front is usually an irregular set made up of several, distinct smooth parts; each of these nice portions of the front is typically the image of a set of solutions sharing the same structure, i.e., associated with the same support set. The rationale of the proposed algorithm is thus to first define a set of starting solutions; the support sets of these solutions should ideally be diverse and define a subspace where a portion of the Pareto set lies. Then, an adaptation of the front steepest algorithm [17, 33] can be run starting from this initial set of solutions to span the front exhaustively. To the best of our knowledge, SFSD is the first front-oriented approach for cardinality-constrained MOO.

4.2.1 Phase One: Initialization

The first phase of the SFSD procedure deals with the identification of a set of starting solutions. The most direct way of proceeding would arguably be exhaustive enumeration of the super support sets, selecting for each a solution. However, the number of

possible supports is high, growing as fast as $\binom{n}{s}$, but only a small fraction contributes to the Pareto front. Thus, this strategy is inefficient, up to being totally impractical with problems of nontrivial size.

A totally random initialization might also appear to be a possible path to take, but, by similar reasons as above, it would end up being a completely luck-based operation. Therefore, we suggest to exploit single-point solvers to retrieve Pareto stationary solutions. Indeed, by the mechanisms of this kind of algorithms, not only the obtained points are stationary but are usually also good solutions from a global optimization perspective. We explored the following (not exhaustive) list of options:

- Using the MOIHT discussed in Sect. 4.1 in a multi-start fashion. Since the algorithm finds L -stationary solutions, optimization should be driven avoiding “bad” supports.
- Using the multi-objective sparse penalty decomposition (MOSPD) method from [31] in a multi-start fashion; in brief, at each iteration k of MOSPD, a pair (x_{k+1}, y_{k+1}) is found such that x_{k+1} is (approximately) Pareto stationary for the penalty function $Q_{\tau_k}(x, y_{k+1}) = F(x) + \frac{\tau_k}{2} \mathbf{1}_m \|x - y_{k+1}\|^2$, with $\tau_k \rightarrow \infty$ for $k \rightarrow \infty$. The pair (x_{k+1}, y_{k+1}) is obtained by means of an *alternate minimization* scheme. For further details, we refer the reader to [31]. MOSPD is proved to converge to points satisfying the multi-objective Lu–Zhang conditions for problem (1), that are even weaker than Pareto stationarity; however, penalty decomposition methods have been shown to retrieve solutions both in the scalar [30] and in the multi-objective [31] case that are good from a global optimization perspective.
- Combining the strategies of the two preceding points: for each point of a multi-start random initialization, we can first run the MOSPD procedure to exploit its exploration capabilities; then, we can use MOIHT in cascade, so that bad Lu–Zhang points that are not L -stationary can eventually be escaped. We refer to this approach as MOHyb.
- Solving the scalarized—single-objective—problem for different trade-off parameters.

Once the starting set of solutions is obtained by one of the above strategies, a further step has to be carried out. Indeed, we need to associate each solution with a super support set. Now, if a solution has full support, then there is a unique super support and no ambiguity. However, there might be solutions with incomplete support; these solutions might be not Pareto stationary (for example, if obtained with MOSPD), in which case we shall carry out a descent step along the steepest feasible descent direction; if, on the other hand, we actually have a Pareto stationary point with incomplete support, we shall associate it with any of the super supports.

Obviously, we can complete this first phase with a filtering operation, where dominated solutions get discarded. To sum up, the result of the first phase of the algorithm provides a set of starting solutions each one associated with a super support set.

4.2.2 Phase Two: Front steepest descent

In Algorithm 2, we report the scheme of the proposed algorithmic framework (SFSD).

Algorithm 2: Sparse Front Steepest Descent

```

1 Input:  $F : \mathbb{R}^n \rightarrow \mathbb{R}^m, \alpha_0 > 0, \delta \in (0, 1), \gamma \in (0, 1)$ .
2  $\mathcal{X}^0 = \text{Initialize}(F)$ 
3  $k = 0$ 
4 while a stopping criterion is not satisfied do
5    $\widehat{\mathcal{X}}^k = \mathcal{X}^k$ 
6   forall the  $(x_c, J_{x_c}) \in \mathcal{X}^k$  do
7     if  $(x_c, J_{x_c}) \in \widehat{\mathcal{X}}^k$  then
8       if  $\theta_{J_{x_c}}(x_c) < 0$  then
9          $\alpha_c^k = \max_{h \in \mathbb{N}} \{ \alpha_0 \delta^h \mid F(x_c + \alpha_0 \delta^h d_{J_{x_c}}(x_c)) \leq F(x_c) + \mathbf{1}_m \gamma \alpha_0 \delta^h \theta_{J_{x_c}}(x_c) \}$ 
10        else
11           $\alpha_c^k = 0$ 
12           $z_c^k = x_c + \alpha_c^k d_{J_{x_c}}(x_c)$ 
13           $\widehat{\mathcal{X}}^k = \widehat{\mathcal{X}}^k \setminus \{ (y, J_y) \in \widehat{\mathcal{X}}^k \mid J_y = J_{x_c}, F(z_c^k) \not\leq F(y) \} \cup \{ (z_c^k, J_{x_c}) \}$ 
14          forall the  $I \subseteq \{1, \dots, m\}$  s.t.  $\theta_{J_{x_c}}^I(z_c^k) < 0$  do
15            if  $(z_c^k, J_{x_c}) \in \widehat{\mathcal{X}}^k$  then
16               $\alpha_c^I = \max_{h \in \mathbb{N}} \{ \alpha_0 \delta^h \mid \forall (y, J_y) \in \widehat{\mathcal{X}}^k, J_y = J_{x_c}, \exists j \in$ 
17                 $\{1, \dots, m\}$  s.t.  $f_j(z_c^k + \alpha_0 \delta^h d_{J_{x_c}}^I(z_c^k)) < f_j(y) \}$ 
18               $\hat{z} = z_c^k + \alpha_c^I d_{J_{x_c}}^I(z_c^k)$ 
19               $\widehat{\mathcal{X}}^k = \widehat{\mathcal{X}}^k \setminus \{ (y, J_y) \in \widehat{\mathcal{X}}^k \mid J_y = J_{x_c}, F(\hat{z}) \not\leq F(y) \} \cup \{ (\hat{z}, J_{x_c}) \}$ 
19    $\mathcal{X}^{k+1} = \widehat{\mathcal{X}}^k$ 
20    $k = k + 1$ 
21 return  $\mathcal{X}^k$ 

```

The method starts working with the starting set of solutions resulting from the Initialize step, i.e., phase one of the algorithm; the obtained set \mathcal{X}^0 is then given by

$$\mathcal{X}^0 = \{ (x, J_x) \mid J_x \in \mathcal{J}(x) \},$$

i.e., solutions associated with a corresponding super support set. Given any pairs $(x, J_x), (y, J_y) \in \mathcal{X}^0$ with $J_x = J_y$, we assume that x and y are mutually nondominated w.r.t. F .

Basically, the SFSD algorithm employs the instructions of the front steepest descent algorithm [17], modified as suggested in [33], treating separately points associated with different super support sets.

Specifically, for any nondominated point x_c in the current Pareto front approximation, a common descent step in the subspace corresponding to the support J_{x_c} is carried out, doing a standard Armijo-type line search [27]. In other words, the search direction is thus given by $d_{J_{x_c}}(x_c)$ according to (5). Then, further searches w.r.t. subsets of objectives are carried out from the obtained point, as long as it is not dominated by any other points y in the set with $J_y = J_{x_c}$. These additional explorations are

carried out along partial descent directions [16, 17] in the reference subspace of the point at hand. Considering $I \subseteq \{1, \dots, m\}$ as a subset of objectives indices, we define $\theta_J^I(\bar{x}) = \min_{d \in \mathbb{R}^n} \max_{j \in I} \nabla f_j(\bar{x})^\top d + \frac{1}{2} \|d\|^2$ s.t. $d_{\bar{J}} = \mathbf{0}_{|\bar{J}|}$. Similar to (5), the problem has a unique solution that we denote by $d_J^I(\bar{x})$.

Since the solutions are compared only if associated with the same super support set, the subspaces induced by different super support sets are explored separately in SFSD. As a result, we basically obtain separate Pareto front approximations, each one corresponding to a super support set. At the end of the SFSD execution, all the points can be compared and the dominated ones can finally be filtered out in order to obtain the final Pareto front approximation for problem (1).

Note that, conceptually, the algorithm can be seen as if multiple, independent runs of the front steepest descent method were carried out, each time constraining the optimization process in a particular subspace; however, exploration in SFSD is carried out “in parallel” throughout different supports, so that the front approximation is constructed uniformly and we can avoid cases where all the computational budget is spent for the optimization w.r.t. the first few considered supports.

Remark 4.4 Since each point is considered for search steps only in the subspace induced by its associated super support set, it easily follows that every new solution will be feasible for (1).

4.2.3 Algorithm Theoretical Analysis

In this section, we state the convergence property of the SFSD methodology. We refer the reader to [33] for the proofs of properties inherited by SFSD directly from the front steepest descent method.

Before proving the convergence result, we need to introduce the set $X_J^k = \{x \mid \exists (x, J) \in \mathcal{X}^k\}$, with J denoting a super support set, and to recall the definition of *linked sequences*, firstly introduced in [37].

Definition 4.1 Let $\{X_J^k\}$ be the sequence of sets of nondominated points, associated with the super support set J , produced by Algorithm 2. We define a linked sequence as a sequence $\{x_{jk}^J\}$ such that, for all k , the point $x_{jk}^J \in X_J^k$ is generated at iteration $k - 1$ of Algorithm 2 by the point $x_{jk-1}^J \in X_J^{k-1}$.

Remark 4.5 The SFSD methodology also inherits the well-definiteness property from the front steepest descent method. In particular, the soundness of the line searches holds by Propositions 3.1, 3.2 and 3.3 stated in [33]. In fact, proofs can be adapted easily, taking into account that SFSD deals separately with one or multiple sets, each corresponding to a different super support set J .

Proposition 4.2 Let us assume that X_J^0 is a set of mutually nondominated points and there exists $x_0^J \in X_J^0$ such that the set $\widehat{\mathcal{L}}_F(x_0^J) = \bigcup_{j=1}^m \{x \in \Omega \mid f_j(x) \leq f_j(x_0^J)\}$ is compact. Moreover, let $\{X_J^k\}$ be the sequence of sets of nondominated points, associated with the super support set J , produced by Algorithm 2, and $\{x_{jk}^J\}$ be a linked

sequence. Then, the latter admits accumulation points, each one satisfying the MOLZ conditions for problem (1).

Proof By the instructions of the algorithm, each linked sequence $\{x_{j_k}^J\}$ can be seen as a linked sequence generated by applying the front steepest descent algorithm from [33] to the problem of minimizing $F(x)$ subject to $x_{\bar{j}} = 0$. Thus, we can follow the proof of [33, Proposition 3.4] to show that each accumulation point \bar{x} of the linked sequence $\{x_{j_k}^J\}$ is such that $\theta_J(\bar{x}) = 0$, i.e., \bar{x} satisfies the MOLZ conditions for (1). \square

5 Computational Experiments

In this section, we report the results of some computational experiments aimed at assessing the numerical potential of the proposed approaches. The code¹ for the experiments, which was written in Python3, was run on a computer with the following characteristics: Ubuntu 22.04, Intel Xeon Processor E5-2430 v2 6 cores 2.50 GHz, 16 GB RAM. In order to solve instances of problems like (4)–(5)–(13), the Gurobi optimizer (version 9.1) was employed.

5.1 Experiments Setup

In our numerical experience, we considered two classes of problems: cardinality-constrained quadratic problems and sparse logistic regression tasks.

The quadratic MOO problems, which often represent a useful test benchmark in optimization, have the form

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \left(x^\top Q_1 x - c_1^\top x, x^\top Q_2 x - c_2^\top x \right)^\top \quad \text{s.t.} \quad \|x\|_0 \leq s,$$

where $Q_1, Q_2 \in \mathbb{R}^{n \times n}$ are random positive semi-definite matrices and $c_1, c_2 \in \mathbb{R}^n$ are vectors whose values are randomly sampled in the range $[-1, 1)$. In the experiments, we varied the following problem parameters: the size $n \in \{10, 25, 50\}$; the condition number of the matrices $\kappa \in \{1, 10, 100\}$; and the cardinality upper bound s . In particular, the latter was set in the following way: for $n = 10$, $s \in \{2, 5, 8\}$; for $n = 25$, $s \in \{5, 10, 20\}$; and for $n = 50$, $s \in \{5, 15, 30\}$. Moreover, we used 3 different seeds for the pseudorandom number generator, thus leading to a total of 81 quadratic problems. For each instance, Q_1 and Q_2 are characterized by the same condition number, i.e., $L(f_1) = L(f_2) = \kappa$.

As for the sparse logistic regression problem [5, 14], it is a relevant task in machine and statistical learning. Given a dataset of N samples with n features $R = (r_1, \dots, r_N)^\top \in \mathbb{R}^{N \times n}$ and N corresponding labels $\{t_1, \dots, t_N\}$ belonging

¹ The implementation code of the methodologies proposed in this paper can be found at <https://github.com/pierlumanzu/cc-moo> [40].

to $\{-1, 1\}$, the *regularized sparse logistic regression problem* is given by:

$$\min_{w \in \mathbb{R}^n} \frac{1}{N} \sum_{i=1}^N \log \left(1 + \exp \left(-t_i \left(w^\top r_i \right) \right) \right) + \frac{\lambda}{2} \|w\|^2 \quad \text{s.t.} \quad \|w\|_0 \leq s,$$

where $\lambda \geq 0$. The logistic loss aims to fit the training data, while the regularization term helps to avoid overfitting. The two functions are clearly in contrast to each other. For our experiments, we employed the multi-objective reformulation considered in [31]:

$$\min_{w \in \mathbb{R}^n} \left(\frac{1}{N} \sum_{i=1}^N \log(1 + \exp(-t_i(w^\top r_i))), \frac{1}{2} \|w\|^2 \right)^\top \quad \text{s.t.} \quad \|w\|_0 \leq s.$$

For this problem, $L(F) = (\|R^\top R\| / N, 1)^\top$. The dataset suite we considered is composed of 7 binary classification datasets from the UCI Machine Learning Repository [23] (Table 1). We tested the algorithms on instances of the problem with $s \in \{2, 5, 8, 12, 20\}$. For each dataset, the samples with missing values were removed. Moreover, the categorical variables were one-hot encoded, while the other ones were standardized to zero mean and unit standard deviation.

In order to evaluate the performance of the algorithms one compared to the others, we employed the performance profiles [21]. In brief, performance profiles show the probability that a metric value achieved by a method in a problem is within a factor $\tau \in \mathbb{R}$ of the best value obtained by any of the solvers considered in the comparison. We refer the reader to [21] for more details. As performance metrics, we used some classical ones of the multi-objective literature: *purity*, Γ -*spread*, Δ -*spread* [19] and *hyper-volume* [49]. Note that, since *purity* and *hyper-volume* have increasing values for better solutions, the performance profiles w.r.t. them were produced based on the inverse of the obtained values.

Note that, for both classes of problems, in the following we will also consider solution approaches based on scalarization, i.e., tackling the problem $\min_{x \in \Omega} f_1(x) + \lambda f_2(x)$, where $\lambda \geq 0$. In the quadratic case, the problem can be solved by means of commercial solvers such as Gurobi, exploiting an MIQP reformulation. In the logistic regression case, we instead use the *Greedy Sparse-Simplex* (GSS) algorithm [1]. Note that, opposed to MIQP approach in quadratic problems, GSS is not guaranteed to produce a Pareto optimal solution.

As anticipated in Sect. 4.2.1, our SFSD methodology was tested taking as starting solutions the ones generated by the single-point methods mentioned above, i.e., MOIHT, MOSPD, MOHYb, MIQP and GSS. Similarly to what is done in [33], we employed a strategy to limit the number of points used for partial descent searches, in order to improve the SFSD efficiency and prevent the production of too many, very close solutions. In detail, we added a condition based on the crowding distance [20] to determine whether a point should be considered for further exploration after the common descent step or not.

Every execution had a time limit of 4 min. In particular, each single-point method was tested in a multi-start fashion: It had to process as many input points as possible

Table 1 Datasets used for the experiments on sparse logistic regression

Dataset	N	n
Heart (Statlog)	270	25
Breast Cancer Wisconsin (prognostic)	194	33
QSAR Biodegradation	1055	41
SPECTF heart	267	44
Spambase	4601	57
Optical recognition of handwritten digits	3823	62
Madelon	2000	500

The number of features takes into account one-hot encoding of categorical ones

within 2 min; in the remaining time, the MOSD procedure was employed as a refiner, starting at each returned point and keeping fixed its zero variables so that the cardinality constraint was kept valid. In SFSD, we set a time limit of 2 min for both phases of the algorithm. For MOIHT, MOSPD and MOHyb, we considered $2n$ initial solutions randomly sampled from a box $([-2, 2]^n$ for the quadratic problems; $[0, 1]^n$ for logistic regression). In order to be feasible, each initial point is first projected onto Ω . These algorithms were executed 5 times with different seeds for the pseudorandom number generator to reduce the sensibility from the random initialization. The five generated fronts were then compared based on the *purity* metric and only the best and worst ones were chosen for the comparisons. The scalarization-based approaches were run once considering $2n$ values for λ , i.e., $\lambda \in \{2^{i+\frac{1}{2}} \mid i \in \mathbb{Z}, i \in [-n, n)\}$, and starting at the initial solution $\mathbf{0}_n \in \Omega$.

5.2 Quadratic Problems

In this section, we report the results on the cardinality-constrained quadratic problems. As for the algorithms parameters, based on some preliminary experiments not reported here for the sake of brevity, we set: $\varepsilon = 10^{-7}$, $L = 1.1\kappa$ for MOIHT; $\tau_{k+1} = 1.5\tau_k$, $\varepsilon_0 = 10^{-2}$, $\varepsilon_{k+1} = 0.9\varepsilon_k$ and $\|x_{k+1} - y_{k+1}\| \leq 10^{-3}$ as stopping condition for MOSPD; the Pareto stationarity approximation degree $\varepsilon = 10^{-7}$ for MOSD; $\alpha_0 = 1$, $\delta = 0.5$ and $\gamma = 10^{-4}$ for all Armijo-type line searches. Possible values for the MOSPD parameter τ_0 are discussed in the next section. The parameters choices for MOIHT and MOSPD were also used in MOHyb.

5.2.1 Preliminary Assessment of MOIHT, MOSPD and MOHyb

We start analyzing the effectiveness of MOIHT, MOSPD and MOHyb, comparing them in Fig. 4 on a selection of quadratic problems. In order to show the differences among the algorithms as clearly as possible, only for this experiment, we considered a single run where the methods took as input the same 25 randomly extracted initial points. Moreover, we set no time limit, so that all the algorithms could process each initial solution until the respective stopping criteria were met.

The MOSPD and MOHyb performance was investigated for values for $\tau_0 \in \{1, 100\}$. (Results for $\tau_0 = 100$ are shown in the left column of the figure, $\tau_0 = 1$ on the right.) The black dots indicate the reference front: The latter is obtained combining the fronts retrieved by running SFSD with all the proposed initialization strategies and discarding the dominated solutions.

In well-conditioned problems ($\kappa = 1$), the MOIHT algorithm performs well, reaching solutions that belong to the reference front. As for MOSPD, the results with $\tau_0 = 100$ are worse, with MOSPD obtaining solutions far from the reference front. The situation is further stressed as the problem dimension n grows. This sounds reasonable, since setting $\tau_0 = 100$ in penalty decomposition schemes binds the variables close to the initial feasible solution and, as a consequence, prevents from exploiting the exploration capabilities of MOSPD. A better choice for τ_0 ($\tau_0 = 1$) improves the performance of the algorithm, although MOIHT still performs better. This result is somewhat in line with the theory: MOIHT generates L -stationary points, whereas MOSPD converges to solutions only guaranteed to satisfy the (weaker) MOLZ conditions. In this scenario, MOHyb inherits the effectiveness of MOIHT: Regardless of the value for τ_0 , it succeeded in getting solutions of the reference front.

In ill-conditioned problems ($\kappa > 1$), the MOIHT performance gets worse: the method struggled in reaching the reference front. This might be explained by the larger values of L that have to be used with these problems ($L = 1.1\kappa$). As the value of L grows, the L -stationarity condition does not provide enough information on the quality of the solution support set. As a consequence, MOIHT can end up in many L -stationary points with “bad” support, i.e., far from the actual Pareto front of the problem. MOSPD with $\tau_0 = 1$ obtained better solutions in these cases. Employing lower values for τ_0 , the approach is initially allowed to search for a good point minimizing F : This feature can be crucial to avoid a large portion of “bad” L -stationary points and to reach solutions in the reference front. MOHyb ($\tau_0 = 1$) proved to be effective in these scenarios too. The hybrid approach, in these ill-conditioned cases, profited from the exploration capabilities of MOSPD, reaching the same solutions. However, like in the well-conditioned case, MOHyb also proved to be less sensitive than MOSPD w.r.t. the value of τ_0 , taking advantage of the MOIHT mechanisms to reach at least L -stationary solutions when $\tau_0 = 100$.

5.2.2 Evaluation of the SFSD Methodology

We start the analysis of the SFSD algorithm performance on the quadratic problems through Fig. 5, where we show how the front descent phase allows to improve the results of basic multi-start approaches corresponding to phase one, i.e., MOIHT, MOSPD, MOHyb and MIQP. According to the results of Sect. 5.2.1, for MOSPD and MOHyb, we set $\tau_0 = 1$.

As anticipated in Sect. 4.2, in cardinality-constrained MOO the Pareto fronts are typically irregular and made up of several smooth parts. The plots in the figure perfectly reflect these characteristics: Each front portion can indeed be associated with a specific support set. Starting from the solutions generated by the single-point methods, the SFSD methodology proves to be effective in exhaustively spanning each portion associated with a support set. This feature allowed our novel front-oriented approach

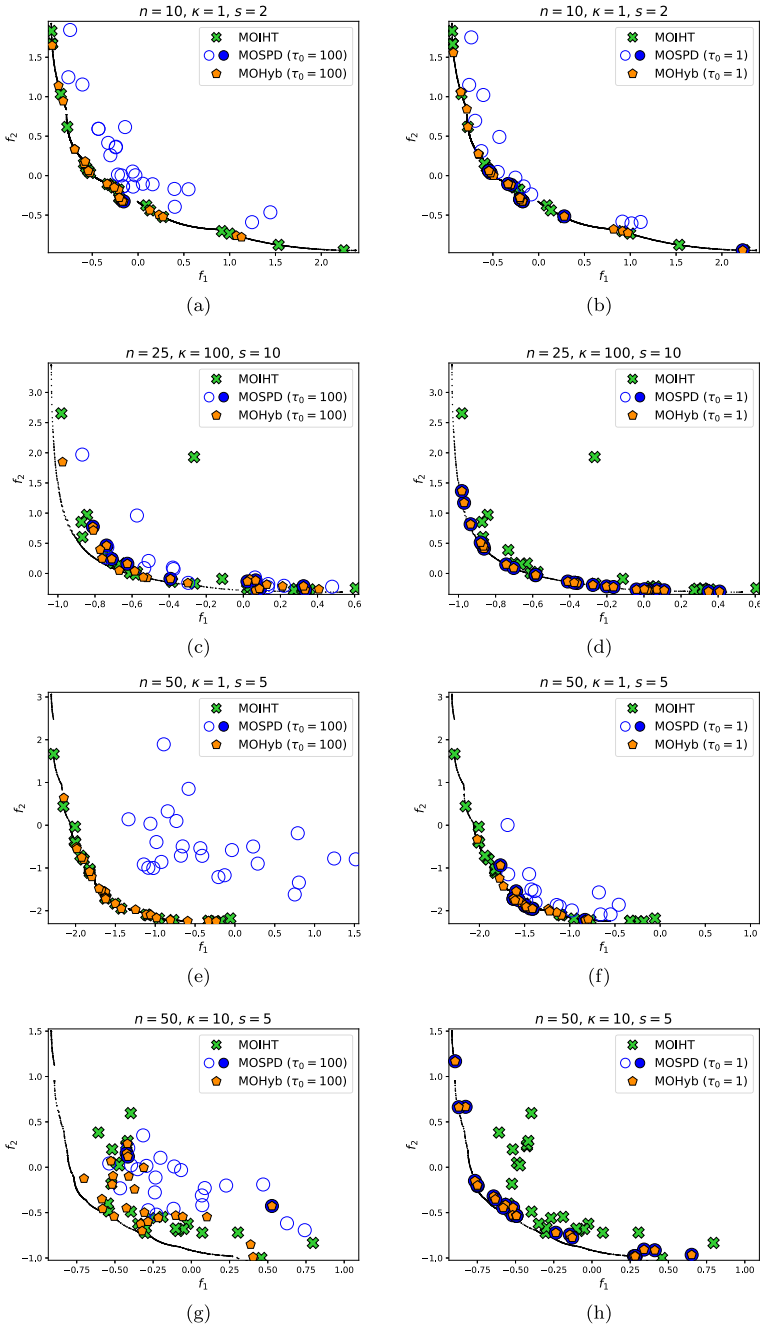


Fig. 4 Results achieved by MOIHT, MOSPD and MOHyb with $\tau_0 \in \{1, 100\}$, starting at 25 random initial solutions, on a selection of quadratic problems. The filled markers denote L -stationary solutions ($L = 1.1\kappa$). The small black dots form the reference front

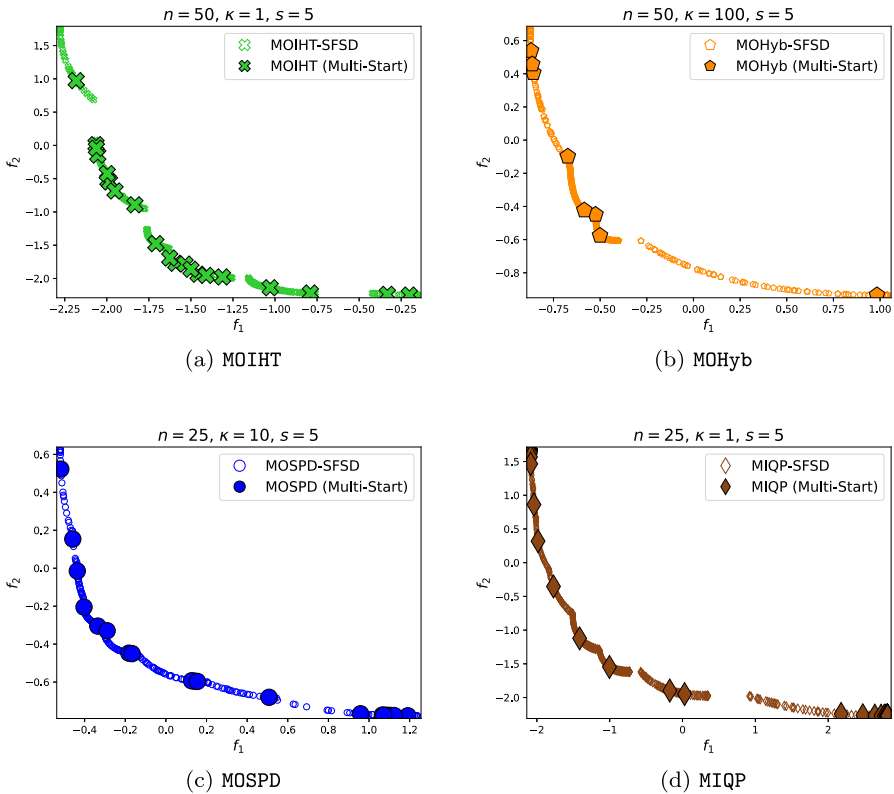


Fig. 5 Results of SFSD phase two compared to simple MOSD refinement of solutions retrieved in phase one. We show one example instance for each considered multi-start/phase one strategy

to identify regions of the Pareto front that would have otherwise be hardly covered with the multi-start strategy.

As mentioned in Sect. 4.2, to the best of our knowledge, SFSD is the first front-oriented approach for cardinality-constrained MOO. In the absence of other specialized algorithms, it is difficult to quantitatively assess the potential of our algorithm and we need to resort to the visual inspection of the solutions. In the rest of the section, we then focus our attention on the different options we outlined for the phase one of the SFSD algorithm, comparing its performance as the initialization strategy varies. The comparisons were made by means of the performance profiles (Fig. 6) on the entire benchmark of quadratic problems.

Looking at the *purity* and the *hyper-volume* metrics, SFSD resulted to be more robust with MOHyb as initialization strategy instead of MOIHT and MOSPD. These results reflect the behavior of the three single-point algorithms already shown in Fig. 4: while MOIHT resulted to be more effective on the well-conditioned problems, MOSPD, with a right choice for the value for τ_0 , performed better on the (larger) set of ill-conditioned problems; MOHyb, inheriting the mechanisms of both, managed to obtain good results on problems of both types. As for Γ -spread, MIQP proves to be more

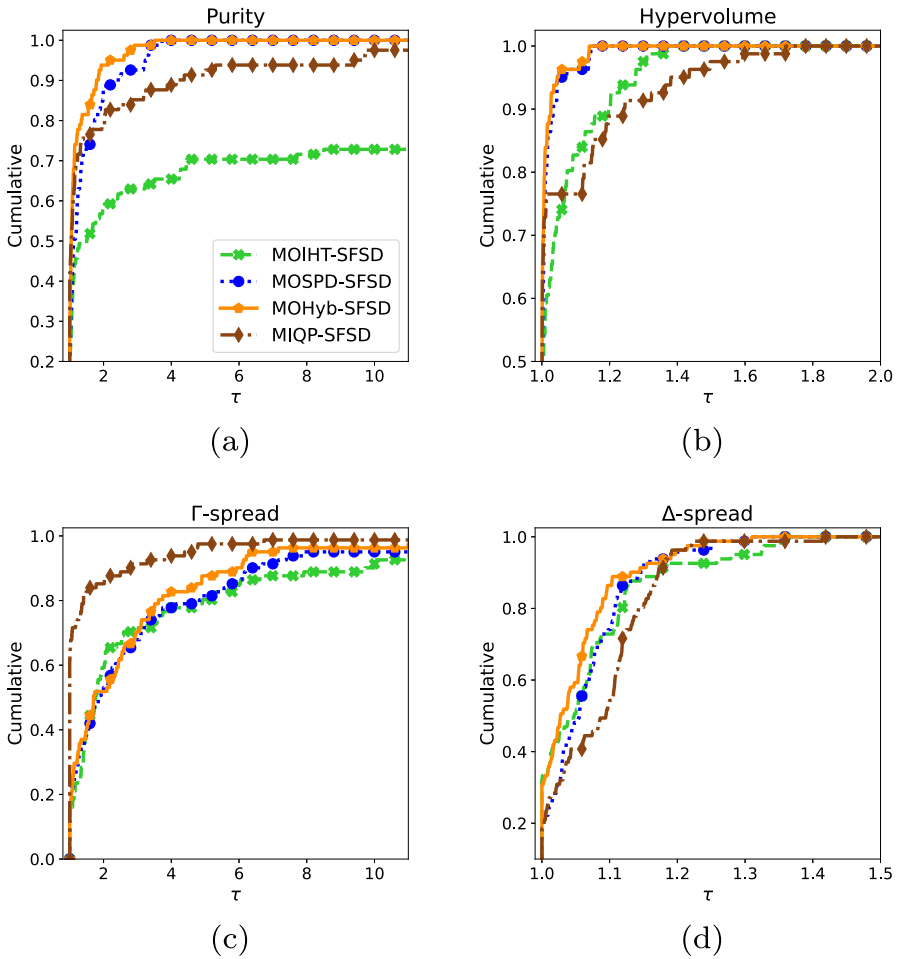


Fig. 6 Performance profiles for SFSD with different initialization strategies, i.e., MOIHT, MOSPD, MOHyb (best executions w.r.t. *purity*) and MIQP on the quadratic problems

capable than the other single-point methods in generating solutions in the extreme regions of the objectives space, and this fact allowed SFSD to get wider front reconstructions. The performance of our front-oriented approach with MOHyb, MOIHT and MOSPD employed in the phase one was quite similar in this scenario. Regarding the Δ -spread metric, i.e., uniformity of the Pareto front approximation, all the initialization strategies led to comparable results.

5.3 Logistic Regression

In this last section, we analyze the performance profiles (Fig. 7) on the logistic regression problems for SFSD with the different possible choices for the first phase of the algorithm. The values for the parameters of the algorithms were again chosen based

on preliminary experiments which are not reported for the sake of brevity. In particular, we set: $L = 1.1 \max\{L(f_1), L(f_2)\}$ for MOIHT; $\varepsilon = 10^{-7}$ for both MOIHT and MOSD; $\tau_0 = 1$, $\tau_{k+1} = 1.3\tau_k$, $\varepsilon_0 = 10^{-5}$, $\varepsilon_{k+1} = 0.9\varepsilon_k$ and $\|x_{k+1} - y_{k+1}\| \leq 10^{-3}$ as stopping condition for MOSPD; $\alpha_0 = 1$, $\delta = 0.5$ and $\gamma = 10^{-4}$ for all Armijo-type line searches. Again, the parameters for MOIHT and MOSPD were also employed in MOHyb. Finally, since the objective functions have different scales, similarly to what is done in [31], when computing the *spread* metrics we considered the logarithm (base 10) of the f_2 values and, then, rescaled both f_1 and f_2 to have values in $[0, 1]$.

With respect to the *purity* and the *hyper-volume* metrics, SFSD resulted to be more robust with MOIHT, MOSPD and MOHyb as initialization strategies, with MOHyb appearing to be slightly superior. As for the Γ -*spread* metric, GSS was the best algorithm for the SFSD phase one. Although SFSD, equipped with this setting, was effective in reaching remote regions of the objectives space, it struggled to obtain uniform Pareto front approximations and, thus, to obtain good Δ -*spread* values. As for this last metric, using as initialization strategy MOIHT proved to be a better choice.

Remark 5.1 In the previous sections, we considered the best executions of SFSD equipped with MOIHT, MOSPD and MOHyb, and we compared them with the deterministic outputs of our front-oriented algorithm when MIQP/GSS was employed in the phase one. Thus, for the sake of completeness, in Fig. 8 we report the performance profiles w.r.t. the *purity* metric obtained considering the worst runs. Comparing these performance profiles with the ones in Figs. 6a and 7a, we observe only slight decreases in the performance of the nondeterministic strategies.

6 Conclusions

In this paper, we considered cardinality-constrained multi-objective optimization problems. Inspired by the homonymous condition for sparse SOO [1], we defined the L -stationarity concept for MOO and we analyzed its relationships with the main Pareto optimality concepts and conditions.

Then, we proposed two novel algorithms for the considered class of problems. The first one is an extension of the *iterative hard thresholding* method [1] to the MOO case, called MOIHT: Like the original approach, it aims to generate an L -stationary point. The second algorithm called *sparse front steepest descent* (SFSD) is, to the best of our knowledge, the first front-oriented approach for cardinality-constrained MOO. Being an adaptation of the front steepest algorithm [17, 33], SFSD aims to approximate the (typically irregular and fragmented) Pareto front of the problem at hand. The method depends on suitable initialization strategies, including, e.g., multi-starting the MOIHT or the MOSPD [31] algorithms, an hybridization of the two, or a scalarization approach. From a theoretical point of view, we proved for MOIHT that the sequence of points converges to L -stationary solutions; for SFSD, on the other hand, we stated global convergence to points satisfying the MO Lu–Zhang optimality conditions.

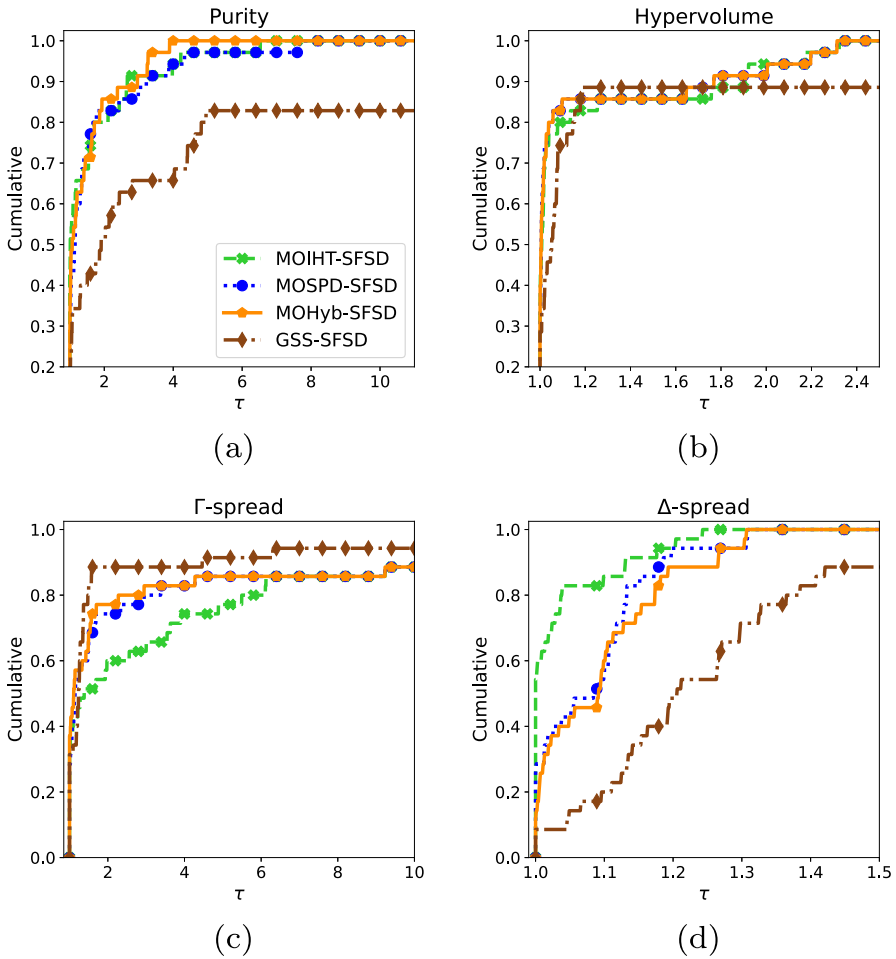


Fig. 7 Performance profiles for SFSD with different initialization strategies, i.e., MOIHT, MOSPD, MOHyb (best executions w.r.t. *purity*) and GSS on 35 logistic regression problems

By a numerical experimentation, we also evaluated the performance of the proposed methodologies on benchmarks of quadratic and logistic regression problems. The SFSD methodology is thus shown to be successful at spanning the Pareto front in an exhaustive way, with the multi-start hybrid MOSPD-MOIHT procedure being the most promising solution to be used in the first phase of the algorithm.

As for future works, we might extend the theoretical results to handle constraints other than the cardinality one. Moreover, further researches might be focused on the performance evaluation of the algorithms on a more extensive set of problems with more general and possibly nonconvex objective functions.

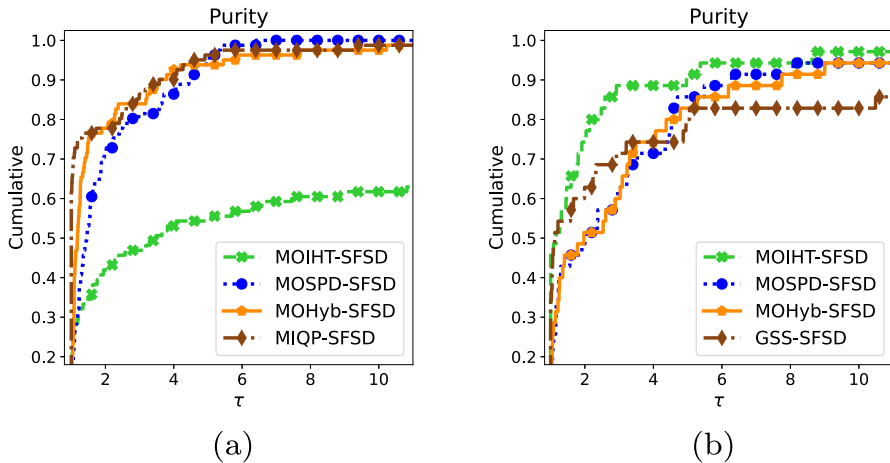


Fig. 8 Performance profiles for SFSD with different initialization strategies, i.e., MOIHT, MOSPD, MOHyb (worst executions w.r.t. *purity*) and MIQP/GSS. **a** Quadratic problems; **b** logistic regression problems

Acknowledgements The authors would like to thank the editor and the anonymous referees for their constructive comments that led to significant improvements to our manuscript.

Funding Open access funding provided by Università degli Studi di Firenze within the CRUI-CARE Agreement. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data Availability All the datasets analyzed during the current study are available in the UCI Machine Learning Repository [23], <https://archive.ics.uci.edu/ml/datasets.php>.

Declarations

Conflicts of interest The authors declare that they have no conflict of interest.

Code Availability Statement The implementation code of the methodologies proposed in this paper can be found at <https://github.com/pierlumanzu/cc-moo> (<https://doi.org/10.5281/zenodo.10473331>).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

A Proof of Lemma 3.2

Proof Since \bar{x} is a Pareto stationarity point for problem (1), by Definition 3.1, we have that $\theta(\bar{x}) = 0$, i.e.,

$$\max_{j \in \{1, \dots, m\}} \nabla f_j(\bar{x})^\top d + \frac{1}{2} \|d\|^2 \geq 0, \quad \forall d \in \mathcal{D}(\bar{x}). \quad (15)$$

Let us suppose, by contradiction, that there exists a direction $\hat{d} \in \mathcal{D}(\bar{x})$ such that

$$\max_{j \in \{1, \dots, m\}} \nabla f_j(\bar{x})^\top \hat{d} < 0. \quad (16)$$

Since (15) holds, we then deduce that $\frac{1}{2} \|\hat{d}\|^2 \geq |\max_{j \in \{1, \dots, m\}} \nabla f_j(\bar{x})^\top \hat{d}| > 0$. Now, let us introduce the function $\tilde{\theta} : \mathbb{R}^n \times \mathbb{R}^n \times [0, 1] \rightarrow \mathbb{R}$ as $\tilde{\theta}(x, d, t) = \max_{j \in \{1, \dots, m\}} \nabla f_j(x)^\top (td) + \frac{1}{2} \|td\|^2 = t \max_{j \in \{1, \dots, m\}} \nabla f_j(x)^\top d + \frac{t^2}{2} \|d\|^2$. By (4) and the feasibility of \hat{d} , it follows that, for all $t \in [0, 1]$, $t\hat{d} \in \mathcal{D}(\bar{x})$ and $\theta(\bar{x}) \leq \tilde{\theta}(\bar{x}, \hat{d}, t)$. It is easy to see that $\tilde{\theta}(\bar{x}, \hat{d}, t) < 0$ if $0 < t < -(2/\|\hat{d}\|^2) \max_{j \in \{1, \dots, m\}} \nabla f_j(\bar{x})^\top \hat{d}$, where the right-hand side is a positive quantity by Equation (16). But, in this case, we would have that $\theta(\bar{x}) \leq \tilde{\theta}(\bar{x}, \hat{d}, t) < 0$, which contradicts the Pareto stationarity of \bar{x} . Thus, we prove that, if \bar{x} is Pareto stationary for problem (1), then $\max_{j \in \{1, \dots, m\}} \nabla f_j(\bar{x})^\top d \geq 0, \forall d \in \mathcal{D}(\bar{x})$. From this point forward, we can follow the proof of Proposition 3.5 in [31]. \square

References

1. Beck, A., Eldar, Y.C.: Sparsity constrained nonlinear optimization: optimality conditions and algorithms. *SIAM J. Optim.* **23**(3), 1480–1509 (2013)
2. Beck, A., Hallak, N.: On the minimization over sparse symmetric sets: projections, optimality conditions, and algorithms. *Math. Oper. Res.* **41**(1), 196–223 (2016)
3. Bertsekas, D.P.: *Nonlinear Programming*, 2nd edn. Athena Scientific, Belmont, MA (1999)
4. Bertsimas, D., Delarue, A., Jaillet, P., Martin, S.: The price of interpretability (2019). arXiv preprint [arXiv:1907.03419](https://arxiv.org/abs/1907.03419)
5. Bertsimas, D., King, A.: Logistic regression: from art to science. *Stat. Sci.* **32**(3), 367–384 (2017)
6. Bertsimas, D., King, A., Mazumder, R.: Best subset selection via a modern optimization lens. *Ann. Stat.* **44**(2), 813–852 (2016)
7. Bertsimas, D., Pauphilet, J., Van Parys, B.: Sparse regression: scalable algorithms and empirical performance. *Stat. Sci.* **35**(4), 555–578 (2020)
8. Bienstock, D.: Computational study of a family of mixed-integer quadratic programming problems. *Math. Program.* **74**(2), 121–140 (1996)
9. Bonnel, H., Iusem, A.N., Svaiter, B.F.: Proximal methods in vector optimization. *SIAM J. Optim.* **15**(4), 953–970 (2005)
10. Burdakov, O.P., Kanzow, C., Schwartz, A.: Mathematical programs with cardinality constraints: reformulation by complementarity-type conditions and a regularization method. *SIAM J. Optim.* **26**(1), 397–425 (2016)
11. Campana, E.F., Diez, M., Liuzzi, G., Lucidi, S., Pellegrini, R., Piccialli, V., Rinaldi, F., Serani, A.: A multi-objective direct algorithm for ship hull optimization. *Comput. Optim. Appl.* **71**(1), 53–72 (2018)
12. Carreira-Perpiñán, M.A., Idelbayev, Y.: “Learning-compression” algorithms for neural net pruning. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018)
13. Carrizosa, E., Frenk, J.B.G.: Dominating sets for convex functions with some applications. *J. Optim. Theory Appl.* **96**(2), 281–295 (1998)
14. Civitelli, E., Lapucci, M., Schoen, F., Sortino, A.: An effective procedure for feature subset selection in logistic regression based on information criteria. *Comput. Optim. Appl.* **80**(1), 1–32 (2021)
15. Cocchi, G., Lapucci, M.: An augmented Lagrangian algorithm for multi-objective optimization. *Comput. Optim. Appl.* **77**(1), 29–56 (2020)

16. Cocchi, G., Lapucci, M., Mansueto, P.: Pareto front approximation through a multi-objective augmented Lagrangian method. *EURO J. Comput. Optim.* **9**, 100008 (2021)
17. Cocchi, G., Liuzzi, G., Lucidi, S., Sciandrone, M.: On the convergence of steepest descent methods for multiobjective optimization. *Comput. Optim. Appl.* **77**(1), 1–27 (2020)
18. 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)
19. 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)
20. 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)
21. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. *Math. Program.* **91**(2), 201–213 (2002)
22. Drummond, L.G., Iusem, A.N.: A projected gradient method for vector optimization problems. *Comput. Optim. Appl.* **28**(1), 5–29 (2004)
23. Dua, D., Graff, C.: UCI Machine Learning Repository (2017). <http://archive.ics.uci.edu/ml>
24. Eichfelder, G.: An adaptive scalarization method in multiobjective optimization. *SIAM J. Optim.* **19**(4), 1694–1718 (2009)
25. Fliege, J.: OLAF—a general modeling system to evaluate and optimize the location of an air polluting facility. *OR-Spektrum* **23**(1), 117–136 (2001)
26. Fliege, J., Drummond, L.M.G., Svaiter, B.F.: Newton’s method for multiobjective optimization. *SIAM J. Optim.* **20**(2), 602–626 (2009)
27. Fliege, J., Svaiter, B.F.: Steepest descent methods for multicriteria optimization. *Math. Methods Oper. Res.* **51**(3), 479–494 (2000)
28. Garmanjani, R., Krulikovski, E., Ramos, A.: On stationarity conditions and constraint qualifications for multiobjective optimization problems with cardinality constraints (2022). https://docentes.fct.unl.pt/algb/files/gkr_mopcac.pdf
29. Jin, Y., Sendhoff, B.: Pareto-based multiobjective machine learning: an overview and case studies. *IEEE Trans. Syst. Man Cybern. Part C (Applications and Reviews)* **38**(3), 397–415 (2008)
30. Kanzow, C., Lapucci, M.: Inexact penalty decomposition methods for optimization problems with geometric constraints. *Comput. Optim. Appl.* **85**(3), 937–971 (2023)
31. Lapucci, M.: A penalty decomposition approach for multi-objective cardinality-constrained optimization problems. *Optim. Methods Softw.* **37**(6), 2157–2189 (2022)
32. Lapucci, M., Levato, T., Rinaldi, F., Sciandrone, M.: A unifying framework for sparsity-constrained optimization. *J. Optim. Theory Appl.* **199**(2), 663–692 (2023)
33. Lapucci, M., Mansueto, P.: Improved front steepest descent for multi-objective optimization. *Oper. Res. Lett.* **51**(3), 242–247 (2023)
34. Lapucci, M., Mansueto, P.: A limited memory quasi-newton approach for multi-objective optimization. *Comput. Optim. Appl.* **85**(1), 33–73 (2023)
35. Lapucci, M., Mansueto, P., Schoen, F.: A memetic procedure for global multi-objective optimization. *Math. Program. Comput.* **15**(2), 227–267 (2022)
36. Laumanns, M., Thiele, L., Deb, K., Zitzler, E.: Combining convergence and diversity in evolutionary multiobjective optimization. *Evol. Comput.* **10**(3), 263–282 (2002)
37. Liuzzi, G., Lucidi, S., Rinaldi, F.: A derivative-free approach to constrained multiobjective nonsmooth optimization. *SIAM J. Optim.* **26**(4), 2744–2774 (2016)
38. Locatelli, M., Maischberger, M., Schoen, F.: Differential evolution methods based on local searches. *Comput. Oper. Res.* **43**, 169–180 (2014)
39. Lu, Z., Zhang, Y.: Sparse approximation via penalty decomposition methods. *SIAM J. Optim.* **23**(4), 2448–2478 (2013)
40. Mansueto, P., Lapucci, M.: MOIHT & SFSD—algorithms for cardinality-constrained multi-objective optimization problems (2024). <https://doi.org/10.5281/zenodo.10473331>
41. Mansueto, P., Schoen, F.: Memetic differential evolution methods for clustering problems. *Pattern Recogn.* **114**, 107–849 (2021)
42. Natarajan, B.K.: Sparse approximate solutions to linear systems. *SIAM J. Comput.* **24**(2), 227–234 (1995)
43. Pascoletti, A., Serafini, P.: Scalarizing vector optimization problems. *J. Optim. Theory Appl.* **42**(4), 499–524 (1984)

44. Pellegrini, R., Campana, E., Diez, M., Serani, A., Rinaldi, F., Fasano, G., Iemma, U., Liuzzi, G., Lucidi, S., Stern, F.: Application of derivative-free multi-objective algorithms to reliability-based robust design optimization of a high-speed catamaran in real ocean environment I. In: *Engineering Optimization IV-Rodrigues*, p. 15 (2014)
45. Tanabe, H., Fukuda, E.H., Yamashita, N.: Proximal gradient methods for multiobjective optimization and their applications. *Comput. Optim. Appl.* **72**(2), 339–361 (2019)
46. Tavana, M.: A subjective assessment of alternative mission architectures for the human exploration of mars at NASA using multicriteria decision making. *Comput. Oper. Res.* **31**(7), 1147–1164 (2004)
47. Tillmann, A.M., Bienstock, D., Lodi, A., Schwartz, A.: Cardinality minimization, constraints, and regularization: a survey (2021). arXiv preprint [arXiv:2106.09606](https://arxiv.org/abs/2106.09606)
48. Weston, J., Elisseeff, A., Schölkopf, B., Tipping, M.: Use of the zero norm with linear models and kernel methods. *J. Mach. Learn. Res.* **3**, 1439–1461 (2003)
49. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms—a comparative case study. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.P. (eds.) *Parallel Problem Solving from Nature—PPSN V*, pp. 292–301. Springer, Berlin (1998)

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