On the Treatment of Optimization Problems with L1 Penalty Terms via Multiobjective Continuation

Katharina Bieker, Bennet Gebken, Sebastian Peitz

On the Treatment of Optimization Problems with L1 Penalty Terms via Multiobjective Continuation

Katharina Bieker, Bennet Gebken and Sebastian Peitz

Abstract—We present a novel algorithm that allows us to gain detailed insight into the effects of sparsity in linear and nonlinear optimization, which is of great importance in many scientific areas such as image and signal processing, medical imaging, compressed sensing, and machine learning (e.g., for the training of neural networks). Sparsity is an important feature to ensure robustness against noisy data, but also to find models that are interpretable and easy to analyze due to the small number of relevant terms. It is common practice to enforce sparsity by adding the ℓ_1-norm as a weighted penalty term. In order to gain a better understanding and to allow for an informed model selection, we directly solve the corresponding multiobjective optimization problem (MOP) that arises when we minimize the main objective and the ℓ_1-norm simultaneously. As this MOP is in general non-convex for nonlinear objectives, the weighting method will fail to provide all optimal compromises. To avoid this issue, we present a continuation method which is specifically tailored to MOPs with two objective functions one of which is the ℓ_1-norm. Our method can be seen as a generalization of well-known homotopy methods for linear regression problems to the nonlinear case. Several numerical examples – including neural network training – demonstrate our theoretical findings and the additional insight that can be gained by this multiobjective approach.

Index Terms—Multiobjective Optimization, Nonsmooth Optimization, Machine Learning, Sparsity

◆

1 INTRODUCTION

Optimization problems are at the center of a very large number of problems in science and engineering. In many situations, the task of minimizing some specified objective function is further complicated by the desire to enforce sparsity, i.e., to obtain a solution with as few non-zero entries as possible, which has several important advantages. Most notably, sparse solutions tend to be robust against noise. Furthermore, they often allow for a better analysis and interpretation due to the small number of relevant terms, for instance in the identification of governing equations [1]. Practical applications can be found in many different areas, e.g., signal processing, compressed sensing, neural network training and medical imaging [2], [3], [4], [5], [6].

In signal and image processing, one frequently considers (overdetermined) linear models with an ℓ_1 penalty term:

$$\min_{x\in\mathbb{R}^n} \|Ax - b\|_2^2 + \lambda \|x\|_1, \tag{1}$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ and $\lambda \in \mathbb{R}^{\geq 0}$. This problem is often referred to as *Lasso* [7] or *basis pursuit* [8]. Since appropriately weighting the penalty term is difficult a priori, the solution is computed for multiple values of λ, for instance in *homotopy methods* [9], [10], [11], [12], where the entire solution path for $\lambda \in \mathbb{R}^{\geq 0}$ is tracked. The method is based on the subdifferential of the ℓ_1-norm and the resulting Karush-Kuhn-Tucker (KKT) conditions, while exploiting the fact that the structure of zero entries along the solution path is locally constant almost everywhere.

- *The authors are with the Department of Mathematics, Paderborn University, Germany.*
  E-mail: bieker@math.upb.de

In recent years, nonlinear models have become increasingly popular, in particular due to the advances in deep learning and the associated neural network training, where sparsity can help to avoid overfitting [13]. As in the linear case, the use of ℓ_1 penalty terms is a natural and common choice for obtaining sparse models, resulting in the problem

$$\min_{x\in\mathbb{R}^n} f(x) + \lambda\|x\|_1 \quad \text{with} \quad \lambda \in \mathbb{R}^{\geq 0}. \tag{2}$$

Since f is based on a nonlinear model and is generally non-convex, (2) is much more difficult to solve than (1).

In contrast to the classical ℓ_1 penalty approach, we will here interpret sparse optimization as a (non-smooth) *multiobjective optimization problem* (MOP), where the objectives f and $\|\cdot\|_1$ are minimized simultaneously, i.e.,

$$\min_{x\in\mathbb{R}^n} \begin{pmatrix} f(x) \\ \|x\|_1 \end{pmatrix}. \tag{MOP-ℓ_1}$$

In this context, (2) is equivalent to the *weighted-sum approach* [14], i.e., $\min_{x\in\mathbb{R}^n} \alpha_1 f(x) + \alpha_2\|x\|_1$ with $\alpha_1, \alpha_2 \in [0,1]$, $\alpha_1 + \alpha_2 = 1$. It is well-known that every solution of the weighted-sum approach is a solution of the MOP [14]. As shown in Figure 1, the weight vector α is orthogonal to the *Pareto front*. Varying α thus yields different compromise solutions. However, if f is non-convex, then α does no longer uniquely define a single Pareto optimal point, and we cannot find points in the non-convex parts of the Pareto front, see [14], [15] for in-depth discussions.

Alternative solution approaches suitable for non-convex problems are *evolutionary algorithms* [16], [17], *scalarization methods* [14], as well as *continuation methods* [18], [19], [20]. The latter require sufficiently smooth objective functions ensuring the existence of a tangent space, in which prediction steps can be performed that are subsequently corrected to obtain another Pareto optimal point.
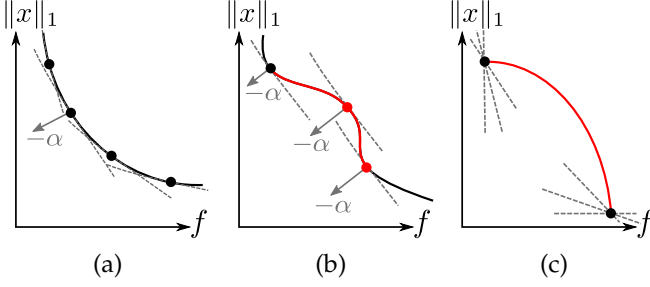
Fig. 1: (a) Different points on the Pareto front by solving (MOP-ℓ_1) using the weighted-sum approach with varying α. (b) For non-convex problems, multiple solutions correspond to the same α such that the solution cannot be computed in the non-convex regions. (c) For strictly concave problems, only the individual minima can be found.

In this paper, we present a new continuation method for non-smooth MOPs with two objectives, where the first objective is twice continuously differentiable and the second objective is the ℓ_1-norm (Section 3). In particular, the first objective may be non-convex. Although the solution set of non-smooth MOPs does not generally possess the smooth structure required for classical continuation, we will show that this specific problem class is piecewise-smooth. Thus, the solution set can be computed via classical continuation up to "kinks". We will show how these kinks can be detected and addressed within the continuation by exploiting the special structure of the optimality conditions specifically derived for this purpose. The global minimal point $0 \in \mathbb{R}^n$ of the ℓ_1-norm can be used as the starting point.

Alternative approaches at the intersection of sparse and multiobjective optimization are, for instance the, training of (linear) SVM models using ε-constraint scalarization [21], [22], or the solution of MOPs with linear models and the ℓ_1-norm via evolutionary algorithms [23]. In the context of sparse optimization, generalizations (for convex problem classes) of the classical homotopy methods were introduced in [24], [25], [26], [27], for instance to arbitrary convex loss-functions or convex penalty terms. Thus, the main advantage of our method is the treatment of nonlinear, non-convex problems, which will be demonstrated using several examples of varying complexity (Section 4).

2 MULTIOBJECTIVE AND NON-SMOOTH OPTIMIZATION

Before presenting the solution approach, we give a short overview of multiobjective and non-smooth optimization. For more details, see [14], [15], [18] and [28], [29], respectively.

The general multiobjective optimization problem is to minimize multiple (real-valued) functions $f_i : \mathbb{R}^n \to \mathbb{R}$, $i \in \{1, ..., k\}$, at the same time:

$$\min_{x \in \mathbb{R}^n} F(x) = \min_{x \in \mathbb{R}^n} \begin{pmatrix} f_1(x) \\ \vdots \\ f_k(x) \end{pmatrix}. \tag{MOP}$$

Due to the loss of a total order of $\mathbb{R}^k$ for $k > 1$, the solution to (MOP) is not a specific point, but the *set of*

non-dominated points, also referred to as the *Pareto set*. Non-dominated points x^* are characterized by the fact that there exists no x with $f_i(x) \leq f_i(x^*)$ for all $i \in \{1, \dots, k\}$ and $f_j(x) < f_j(x^*)$ for at least one $j \in \{1, \dots, k\}$.

If all objectives are continuously differentiable, then these points satisfy the well-known necessary KKT conditions, which state that if a point x^* is locally Pareto optimal, there exists a convex combination of the individual gradients that is zero [18], i.e., there exists $\alpha^* \in (\mathbb{R}^{\geq 0})^k$ with

$$\sum_{i=1}^{k} \alpha_i^* \nabla f_i(x^*) = 0 \text{ and } \sum_{i=1}^{k} \alpha_i^* = 1.$$

Points satisfying the KKT conditions are denoted as *Pareto critical*, and the set containing all Pareto critical points is the *Pareto critical set*, which is a superset of the Pareto set.

As the ℓ_1-norm does not satisfy the required smoothness assumptions for the classical KKT conditions, we need to consider subgradients instead.

Definition 2.1 (Subdifferential and subgradient). *Let $f : \mathbb{R}^n \to \mathbb{R}$ be locally Lipschitz continuous, and let $\Omega \subset \mathbb{R}^n$ be the set of points where f is not differentiable. Then*

$$\partial f(x) := \mathbf{Conv}(\{g \in \mathbb{R}^n \mid \exists (x_j)_j \in \mathbb{R}^n \setminus \Omega \text{ with } x_j \to x \text{ and}$$
$$\nabla f(x_j) \to g \text{ for } j \to \infty\})$$

is the (Clarke) subdifferential *of f in x. An element $g \in \partial f(x)$ is called a* subgradient.

If f is continuously differentiable in x, then the subdifferential is the set containing the gradient, i.e., $\partial f(x) = \{\nabla f(x)\}$. In case of the ℓ_1-norm, the subgradients take the following form.

Example 2.2 (Subgradients and subdifferential of the ℓ_1-norm). *The ℓ_1-norm in $\mathbb{R}^n$, $f_{\ell_1}(x) = \|x\|_1 = \sum_{i=1}^{n} |x_i|$, is non-differentiable in all x with $x_j = 0$ for some $j \in \{1, \dots, n\}$. To calculate $\partial f_{\ell_1}(x)$ for an arbitrary $x \in \mathbb{R}^n$, we first distinguish between* **active** *and* **inactive** *indices, with $\{j \in J \mid x_j \neq 0\} =: A(x)$ being the* **active set**. *It is easy to see that*

$$g = \begin{pmatrix} g_1 \\ \vdots \\ g_n \end{pmatrix} \in \partial f_{\ell_1}(x) \Leftrightarrow \begin{cases} g_j = \mathrm{sgn}(x_j) & \text{if } j \in A(x), \\ g_j \in [-1, 1] & \text{if } j \notin A(x). \end{cases}$$

We denote the number of active indices by $n^A(x) := |A(x)|$ and the number of inactive indices by $n^0(x) := n - n^A(x)$. The subdifferential is now the convex hull of $M(x) := 2^{n^0(x)}$ vectors g^i containing the extreme points of the subdifferential:

$$\partial f_{\ell_1}(x) = \mathbf{Conv}\left(\{g^i \mid i \in \{1, \dots, M(x)\}\}\right),$$

i.e.,

$$g_j^i = \mathrm{sgn}(x_j) \; \forall j \in A(x) \quad \text{and} \quad g_j^i \in \{-1, 1\} \; \forall j \notin A(x).$$

Consider the point $\hat{x} = (2, 0, -3, 0)^\top \in \mathbb{R}^4$ for example. Then we have $A(\hat{x}) = \{1, 3\}$, $n^A(\hat{x}) = 2$, $n^0(\hat{x}) = 2$, $M(\hat{x}) = 4$, and

$$\partial f_{\ell_1}(\hat{x}) = \mathbf{Conv}\left(\left\{\begin{pmatrix} 1 \\ 1 \\ -1 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \\ -1 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ -1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \end{pmatrix}\right\}\right).$$

Analogous to the smooth case, necessary optimality conditions for MOPs containing non-smooth objective functions can be obtained [30].

Proposition 2.3 (Non-smooth KKT conditions). *Assume that $F : \mathbb{R}^n \to \mathbb{R}^k$ is locally Lipschitz continuous and x^* is locally Pareto optimal. Then*

$$0 \in \textbf{Conv} \left(\bigcup_{i=1}^{k} \partial f_i(x^*) \right) \subset \mathbb{R}^n. \qquad (3)$$

Similar to the smooth case, we call $x \in \mathbb{R}^n$ *Pareto critical* if it satisfies the KKT conditions (3) and we call the set of all those points the *Pareto critical set* $\mathcal{P}_c$.

3 THE CONTINUATION METHOD

Our goal is to compute the entire Pareto critical set $\mathcal{P}_c$ of (MOP-ℓ_1) using a continuation method that exploits the structure of the ℓ_1-norm and its subgradients. As a starting point we can use $x = 0$ which is always Pareto critical as the minimum of the ℓ_1-norm, and then move along $\mathcal{P}_c$ in the direction where the value of f decreases and $\|x\|_1$ increases.

Continuation methods (or *predictor-corrector (PC) methods*) yield a sequence of Pareto critical points by – starting from a Pareto critical point – performing a prediction step along the tangent space of $\mathcal{P}_c$ and then a correction step which results in the next Pareto critical point close by, see [18] for details. For the non-smooth problem (MOP-ℓ_1), the Pareto critical set may contain kinks such that we can not exclusively rely on the tangent space in the prediction step. Thus, when we reach such a kink, we additionally require a mechanism that computes all possible directions in which the Pareto critical set might continue, see Algorithm 1 for a sketch. To this end, we will first have a more detailed look at the KKT conditions (3) for the special case of problem (MOP-ℓ_1) in Section 3.1 before giving a detailed description of the predictor and corrector steps in Sections 3.2 and 3.3. We then discuss strategies for the (de-)activation of indices (i.e., how to overcome kinks) in Section 3.4, and subsequently address implementation and globalization in Section 3.5.

3.1 Optimality conditions for (MOP-ℓ_1)

Since we assume f to be at least twice continuously differentiable, the gradient $\nabla f(x)$ exists for every $x \in \mathbb{R}^n$. The

---

Algorithm 1 PC algorithm

Input: $f : \mathbb{R}^n \to \mathbb{R} \in \mathcal{C}^2$, $x_{\text{start}} \in \mathbb{R}^n$ satisfying (3)
Output: Discretization of the Pareto critical set $\mathcal{P}_{\text{PC}}$
$\mathcal{P}_{\text{PC}} = \emptyset$, $x^0 = x_{\text{start}}$
while End of $\mathcal{P}_c$ component has not been reached **do**
  $\mathcal{P}_{\text{PC}} = \mathcal{P}_{\text{PC}} \cup \{x^0\}$
  if $\mathcal{P}_c$ is locally smooth **then**
    Predictor step tangential to $\mathcal{P}_c$: $x^p \leftarrow \text{predictor}(x^0)$
    Corrector step with $x^c \in \mathcal{P}_c$: $x^c \leftarrow \text{corrector}(x^p)$
    $x^0 = x^c$
  else
    (De-)Activate entries
  end if
end while

---

subdifferential of the ℓ_1-norm is given by the convex hull of $M(x) = 2^{n^0(x)}$ vectors g^i given in Example 2.2. To simplify the notation, we will write M and n^0 instead of $M(x)$ and $n^0(x)$. Inserting this into the KKT conditions (3), a point x is Pareto critical for (MOP-ℓ_1) if there exist KKT multipliers $\alpha_1, \alpha_2 \in [0, 1]$ with $\alpha_1 + \alpha_2 = 1$ and

$$\alpha_1 \nabla f(x) + \alpha_2(\beta_1 g^1 + \cdots + \beta_M g^M) = 0, \qquad (4)$$

where $\beta_i \in [0, 1]$ for $i = 1, \ldots, M$ and $\sum_{i=1}^{M} \beta_i = 1$. The following theorem will further simplify this condition.

Theorem 3.1. *Let $x \in \mathbb{R}^n$ with $x \neq 0$ and let $a \in A(x) = \{j \in \{1, \ldots, n\} \mid x_j \neq 0\}$ be any active index. Then, the following statements are equivalent:*

1) *x is Pareto critical for (MOP-ℓ_1)*
2) *The following conditions hold for every $j \in \{1, \ldots, n\}$:*
  a) If $j \in A(x)$:
    i) $\text{sgn}((\nabla f(x))_j) = -\text{sgn}(x_j)$ if $(\nabla f(x))_j \neq 0$ and
    ii) $|(\nabla f(x))_j| = |(\nabla f(x))_a|$
  b) If $j \notin A(x)$: $|(\nabla f(x))_j| \leq |(\nabla f(x))_a|$

Proof. We start by showing $1 \Rightarrow 2$. Since x is Pareto critical, there exist $\alpha_1, \alpha_2 \in [0, 1]$ with $\alpha_1 + \alpha_2 = 1$ and $\beta_i \in [0, 1]$, $i = 1, \ldots, M$, with $\sum_{i=1}^{M} \beta_i = 1$ such that (4) is satisfied.

Note that $\alpha_1 = 0$ would imply that zero is contained in the subdifferential of the ℓ_1-norm at x, which is only the case for $x = 0$. Thus, we must have $\alpha_1 > 0$.

First, assume $j \in A(x)$. Then, $(g^i(x))_j = \text{sgn}(x_j)$ for all $i = 1, \ldots, M$ and

$$0 = \alpha_1(\nabla f(x))_j + \alpha_2 \text{sgn}(x_j)$$
$$= \alpha_1(\nabla f(x))_j + (1 - \alpha_1)\text{sgn}(x_j).$$

Since $\alpha_1 > 0$, this is equivalent to

$$(\nabla f(x))_j = -\frac{(1 - \alpha_1)}{\alpha_1}\text{sgn}(x_j).$$

Therefore, we have

$$\text{sgn}(\nabla f(x))_j = -\text{sgn}(x_j) \quad \Rightarrow \quad |(\nabla f(x))_j| = \frac{(1 - \alpha_1)}{\alpha_1}.$$

Since this has to hold for every active j, it follows

$$|(\nabla f(x))_j| = |(\nabla f(x))_a|.$$

Now assume that j is inactive. Define

$$\beta_{j,-} := \sum_{i:g_j^i=-1} \beta_i, \quad \beta_{j,+} := \sum_{i:g_j^i=1} \beta_i.$$

From (4) we obtain

$$0 = \alpha_1(\nabla f(x))_j + \alpha_2(\beta_{j,+} - \beta_{j,-})$$
$$\Rightarrow \quad |(\nabla f(x))_j| = |\beta_{j,-} - \beta_{j,+}| \frac{1 - \alpha_1}{\alpha_1}$$
$$\leq \frac{1 - \alpha_1}{\alpha_1} = |(\nabla f(x))_a|.$$

For the opposite direction $'' \Leftarrow ''$, assume that x satisfies 2(a) and (b). Note that 2(a) implies $\text{sgn}(x_a)(\nabla f(x))_a < 0$ and thus,

$$1 - \text{sgn}(x_a)(\nabla f(x))_a > 1.$$

Define

$$\alpha_1 := \frac{1}{1 - \text{sgn}(x_a)(\nabla f(x))_a} \in (0, 1).$$

By 2(a)(i) we obtain

$$\alpha_1 = \frac{1}{1 + |(\nabla f(x))_a|}.$$

With 2(a)(i) it follows that for all active j

$$\frac{\alpha_1}{1 - \alpha_1} |(\nabla f(x))_j| = \frac{1}{\frac{1}{\alpha_1} - 1} |(\nabla f(x))_a| = 1.$$

In particular,

$$1 = \frac{\alpha_1}{1 - \alpha_1} |(\nabla f(x))_j| = \frac{\alpha_1}{1 - \alpha_1} \text{sgn}(\nabla f(x)_j)\nabla f(x)_j$$
$$\Rightarrow \quad \frac{\alpha_1}{1 - \alpha_1} \nabla f(x)_j = \text{sgn}(\nabla f(x)_j) = -\text{sgn}(x_j).$$

Analogously, for all inactive j, we can use 2(b) to obtain

$$\frac{\alpha_1}{1 - \alpha_1} |(\nabla f(x))_j| \leq 1 \quad \Rightarrow \quad \frac{\alpha_1}{1 - \alpha_1}(\nabla f(x))_j \in [-1, 1].$$

As a result, the vector $-\frac{\alpha_1}{1-\alpha_1}\nabla f(x)$ can be written as a convex combination of the vectors g^i defined in Example 2.2. Let $\beta_1, ..., \beta_M$ be the corresponding coefficients and define $\alpha_2 = 1 - \alpha_1$. Then

$$-\frac{\alpha_1}{1 - \alpha_1}\nabla f(x) = \beta_1 g^1 + \cdots + \beta_M g^M$$
$$\Leftrightarrow \quad \alpha_1 \nabla f(x) + \alpha_2(\beta_1 g^1 + \cdots + \beta_M g^M) = 0,$$

showing that (4) holds.

$\square$

Theorem 3.1 states that the absolute values of all gradient entries belonging to active indices are identical. At the same time, the absolute values of gradient entries belonging to inactive indices must always be less than or equal to the absolute value of active gradient entries. Furthermore, the proof of the theorem yields an explicit formula for the KKT multipliers of Pareto critical points.

Remark 3.2. *If x is Pareto critical for (MOP-ℓ_1) and a is an active index, then KKT multipliers corresponding to x are given by*

$$\alpha_1 = \frac{1}{1 + |(\nabla f(x))_a|}, \quad \alpha_2 = \frac{|(\nabla f(x))_a|}{1 + |(\nabla f(x))_a|}.$$

If $\nabla f(x) \neq 0$ then the KKT multipliers are unique. In particular, this indicates that "kinks" in the Pareto front can only occur if $\nabla f(x) = 0$.

As mentioned before, since (MOP-ℓ_1) is a non-smooth problem, the Pareto critical set will generally contain points in which the tangent space does not exist. Theorem 3.1 can be used to characterize those points and to gain insight into the structure of $\mathcal{P}_c$ by discriminating between equality and strict inequality in condition 2(b).

Corollary 3.3. *Let $x^0 \neq 0$ be a Pareto critical point such that the inequality in 2(b) of Theorem 3.1 is strict. Let $j_1 < ... < j_{n^A(x^0)}$ be the (sorted) active indices at x^0. Define the following*

two mappings that lift the active components to the n-dimensional point x and vice versa:

$$h^1 : \mathbb{R}^{n^A(x^0)} \to \mathbb{R}^n, \quad h_j^1(\bar{x}) = \begin{cases} 0 & \text{if } j \notin A(x^0), \\ \bar{x}_i & \text{if } j_i \in A(x^0), \end{cases}$$

$$h^2 : \mathbb{R}^n \to \mathbb{R}^{n^A(x^0)}, \quad h_i^2(x) = x_{j_i} \quad \forall i \in \{1, ..., n^A(x^0)\}.$$

Then there is an open set $U \subseteq \mathbb{R}^n$ with $x^0 \in U$ such that $x \in U$ being Pareto critical for (MOP-ℓ_1) is equivalent to $A(x) = A(x^0)$ and $h^2(x)$ being Pareto critical for the MOP

$$\min_{\bar{x} \in \mathbb{R}^{n^A(x^0)}} \begin{pmatrix} f(h^1(\bar{x})) \\ \|\bar{x}\|_1 \end{pmatrix}. \tag{5}$$

Proof. By definition of $A(x^0)$, there has to be some $U^1 \subseteq \mathbb{R}^n$ with $x^0 \in U^1$ such that

$$\text{sgn}(x_j) = \text{sgn}(x_j^0) \neq 0 \quad \forall j \in A(x^0), \ x \in U^1.$$

In particular,

$$A(x^0) \subseteq A(x) \quad \forall x \in U^1. \tag{6}$$

Let $a \in A(x^0)$. Since ∇f is continuous and since the inequality in 2(b) of Theorem 3.1 is strict, there has to be an open set $U^2 \subseteq \mathbb{R}^n$ with $x^0 \in U^2$ such that for all $x \in U^2$

$$|(\nabla f(x))_j| < |(\nabla f(x^0))_a| \ \forall j \notin A(x^0) \quad \text{and}$$
$$\min_{j \notin A(x^0)} |(\nabla f(x))_j| < \min_{j \in A(x^0)} |(\nabla f(x))_j|.$$

This implies

$$A(x) \subseteq A(x^0) \quad \forall x \in U^2 \cap \mathcal{P}_c. \tag{7}$$

Now let $U = U^1 \cap U^2$ and $x \in U$. By (6), (7) and Theorem 3.1, x being Pareto critical for (MOP-ℓ_1) is equivalent to

$$A(x) = A(x^0),$$
$$\text{sgn}((\nabla f(x))_j) = -\text{sgn}(x_j) \quad \forall j \in A(x^0),$$
$$|(\nabla f(x))_j| = |(\nabla f(x))_a| \quad \forall j \in A(x^0).$$

By applying Theorem 3.1 to the MOP (5) (which has the same form as (MOP-ℓ_1)), we see that the last two lines of the previous system of equations are precisely the conditions for $h^2(x)$ being Pareto critical. This completes the proof.

$\square$

The previous corollary implies that if the inequality in 2(b) of Theorem 3.1 is strict for a point $x^0 \in \mathcal{P}_c$, then locally around x^0, the Pareto critical set of (MOP-ℓ_1) coincides with the Pareto critical set of the MOP (5), embedded from $\mathbb{R}^{n^A(x^0)}$ to $\mathbb{R}^n$ via h^1. The objectives of (5) are smooth around $h^2(x^0)$, such that we can apply the theory for smooth MOPs to see that the tangent space of the Pareto critical set $\mathcal{P}_c$ must exist in x^0. Thus, kinks in $\mathcal{P}_c$ can only arise in points where equality holds in 2(b) for some $j \notin A(x^0)$. These are precisely the points where the active set can potentially change, i.e., where an inactive index might be activated (or vice versa).

Corollary 3.3 shows that it is sufficient to consider the MOP (5) if we are in a point where the inequality in 2(b) of Theorem 3.1 is strict. Based on (5), this allows us to construct the predictor (Section 3.2) and corrector (Section 3.3) as in the smooth case except for kinks.

3.2 Predictor

In the predictor step, we perform a step along the tangent space of the Pareto critical set from the starting point $x^0 \in \mathcal{P}_c$. As discussed above, we assume that an open set around x^0 exists in which the zero structure of Pareto critical points does not change, such that the original MOP locally reduces to the problem (5), where both objective functions are at least twice continuously differentiable. This means that we can compute the predictor according to classical continuation methods, cf. [18]. More precisely, let $H : \mathbb{R}^{n^A(x^0)+2} \to \mathbb{R}^{n^A(x^0)+1}$,

$$H(\bar{x}, \alpha) = \begin{pmatrix} \alpha_1 \nabla(f \circ h^1)(\bar{x}) + \alpha_2 \nabla f_{\ell_1}(\bar{x}) \\ \alpha_1 + \alpha_2 - 1 \end{pmatrix}$$

be the map containing the KKT conditions of (5). Then the tangent space of the Pareto critical set is given by the first $n^A(x^0)$ components of

$$\ker(H'(h^2(x^0), \alpha))$$
$$= \ker \begin{pmatrix} \alpha_1 \nabla^2(f \circ h^1)(h^2(x^0)) \\ 0 \end{pmatrix}$$
$$\qquad \begin{pmatrix} \nabla(f \circ h^1)(h^2(x^0)) & \nabla f_{\ell_1}(h^2(x^0)) \\ 1 & 1 \end{pmatrix}$$
$$= \ker \begin{pmatrix} \alpha_1 \nabla^2 f(x^0)_{|A(x^0)} & \nabla f(x^0)_{|A(x^0)} & s \\ 0 & 1 & 1 \end{pmatrix}, \qquad (8)$$

where $(\nabla^2 f(x))_{|A(x^0)}$ and $\nabla f(x)_{|A(x^0)}$ denote the hessian and the gradient of f reduced to the active indices, respectively, and $s = \text{sgn}(h^2(x^0)) = \text{sgn}(x^0)_{|A(x^0)}$ is the result of the componentwise sign function in the active indices of x^0. Note that Theorem 3.1 implies $\nabla f(x^0)_{|A(x^0)} = -\left| (\nabla f(x^0))_a \right| s$.

If the reduced hessian $\nabla^2 f(x)_{|A(x^0)}$ is regular, then the kernel (i.e., the tangent space) is one dimensional and we obtain a unique vector (up to scaling and sign). As discussed in [31], we expect this to be the generic case and assume that this holds for the remainder of this paper. Otherwise, the kernel could be of higher dimension which would lead to infinitely many possible directions. In this case, we could try to find a suitable direction in the kernel by using the directional derivative of the gradient and the fact that the absolute values of the entries of the gradient in the active indices must coincide (cf. 2(a) of Theorem 3.1).

If the reduced hessian $\nabla^2 f(x)_{|A(x^0)}$ is regular, then a kernel vector $\bar{v}$ of (8) is given by

$$\bar{v} = \gamma \cdot (\bar{v}_1, \bar{v}_2, \bar{v}_3)^\top \quad \text{with} \quad \gamma \in \mathbb{R} \quad \text{and}$$
$$\bar{v}_1 = \frac{(1 + |(\nabla f(x^0))_a|)}{\alpha_1} \left(\nabla^2 f(x^0)_{|A(x^0)} \right)^{-1} s,$$
$$= (1 + |(\nabla f(x^0))_a|)^2 \left(\nabla^2 f(x^0)_{|A(x^0)} \right)^{-1} s, \qquad \text{(P)}$$
$$\bar{v}_2 = 1,$$
$$\bar{v}_3 = -1.$$

The predictor of (5) is given by $\bar{v}_1$, while $\bar{v}_2$ and $\bar{v}_3$ represent the corresponding direction in the space of KKT multipliers. The predictor of (MOP-ℓ_1) is then $v_1 = h^1(\bar{v}_1)$.

3.2.1 Direction

If the reduced hessian is regular, there are only two possible choices for the direction of the predictor step, namely v_1

and $-v_1$. As we want to move along the Pareto front in the direction where f becomes smaller (and the ℓ_1-norm larger), it would be intuitive to choose the sign of $\nabla f(x)^\top v_1$ as an indicator, as proposed in [18] and [19]. While this mostly works, the need to cross a "turning point" may arise. These are points where we follow the Pareto critical set in the same direction but the corresponding direction in the image space vanishes and then turns around. This phenomenon is not related to the non-smoothness of the ℓ_1-norm but can also occur in the smooth parts, i.e., on a single *activation structure* (the set of points $x \in \mathcal{P}_c$ with a constant active set $A(x)$), as shown in the following example.

Example 3.4. *Consider* (MOP-ℓ_1) *with*

$$f(x) = (x_1 + 1)^2 + (x_2 - 1)^4 - \frac{1}{2}\left(x_2 - \frac{1}{4}\right)^3. \qquad (9)$$

Figure 2 shows the Pareto critical set for this problem, which can be computed by hand in this case. We see that we have two turning points x^1 and x^2 in the Pareto critical set, where the direction in the image space turns around.
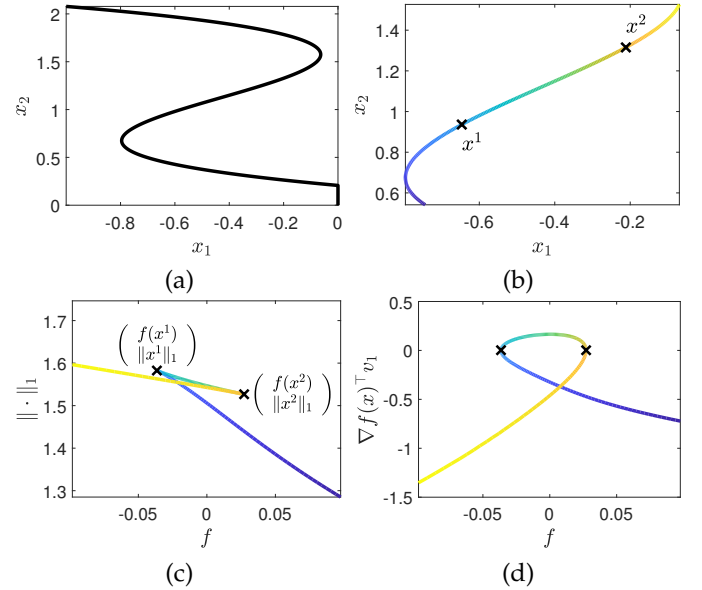

Fig. 2: (a) $\mathcal{P}_c$ for problem (9). (b) Part of $\mathcal{P}_c$ where the turning points x^1 and x^2 occur. (c) Image of $\mathcal{P}_c$ with the same coloring as in (b). (d) Derivative of f in the direction of the predictor, i.e., $\nabla f(x)^\top v_1$.

We thus use a criterion which stems from classical continuation for dynamical systems in [31]. It is based on the sign of the determinant of the augmented jacobian $\left(H'(h^2(x^0), \alpha)^\top, \bar{v} \right)^\top$, which is here given by

$$J_a := \begin{pmatrix} \alpha_1 \nabla^2 f(x^0)_{|A(x^0)} & \nabla f(x^0)_{|A(x^0)} & s \\ 0 & 1 & 1 \\ \bar{v}_1^\top & \bar{v}_2 & \bar{v}_3 \end{pmatrix}. \qquad (10)$$

If the sign of $\det(J_a)$ is constant in one activation structure, then this ensures that $\bar{v}$ has the same orientation with respect to the image of $H'(h^2(x^0), \alpha)$. Furthermore, it prevents us from walking back in the direction we came from (cf. [31]). As we will see in Section 3.4, the correct direction is known at the beginning of a new activation structure.

Therefore, this is a suitable criterion to choose the direction in each predictor step. Furthermore, a simple calculation shows that

$$\det(J_a) = -\bar{v}_2 \cdot z \cdot \det(\alpha_1 \nabla^2 f(x^0)_{|A(x^0)}), \quad (11)$$

where $z := 2 + (1 + |(\nabla f(x^0))_a|)^4 \left\|(\nabla^2 f(x^0)_{|A(x^0)})^{-1} s\right\|_2^2 > 0$ with $a \in A(x^0)$. As a result, only the determinant of $\nabla^2 f(x^0)_{|A(x^0)}$ has to be computed in order to determine the sign of $\det(J_a)$.

3.2.2 Step size

Now it remains to choose a step size $h \in \mathbb{R}^{>0}$ that determines how far we move along the tangent direction, i.e., that determines the predicted point $x^p = x^0 + h \cdot (\pm v_1)$. To obtain an even coverage of the Pareto critical set, we choose a constant step size in the parameter space, i.e.,

$$h := \frac{\tau}{\|v_1\|_2} \quad (12)$$

with a constant $\tau \in \mathbb{R}^{>0}$. For an even distribution in the image space, see [18], [19].

Since we still assume that we are in a part of the Pareto critical set where the activation structure and sign of the variables does not change, the issue may occur that h is too large if we choose it according to one of the above formulas. Therefore, we will choose $\hat{h} = \min(h, \bar{h})$, where $\bar{h}$ determines the maximal length of v_1, such that the activation structure and sign does not change, i.e., such that no active index gets inactive or crosses 0.

3.3 Corrector

The corrector step maps the predicted point onto the Pareto critical set by finding the nearest point $x^c \in \mathcal{P}_c$ with the same zero structure and signs as the predicted point x^p, i.e., with $A(x^c) = A(x^p) = A(x^0)$. The standard approach is to solve the system of equations given by the KKT conditions (4) with x^p as starting point. Since the subgradient of the ℓ_1-norm consists of $M(x^0) = 2^{n^0(x^0)}$ vectors, this can result in very large systems. Fortunately, we can exploit Theorem 3.1 to obtain a simpler way, and solve the following optimization problem:

$$\min_{x \in \mathbb{R}^n} \|x - x^p\|_2^2, \quad s.t.$$
$$\begin{aligned}
(\nabla f(x))_j^2 - (\nabla f(x))_a^2 &= 0 && \text{if } x_j^0 \neq 0 \\
(\nabla f(x))_j^2 - (\nabla f(x))_a^2 &\leq 0 && \text{if } x_j^0 = 0 \\
(\nabla f(x))_j &\leq 0 && \text{if } x_j^0 > 0 \\
(\nabla f(x))_j &\geq 0 && \text{if } x_j^0 < 0 \\
x_j &\geq 0 && \text{if } x_j^0 > 0 \\
x_j &\leq 0 && \text{if } x_j^0 < 0 \\
x_j &= 0 && \text{if } x_j^0 = 0
\end{aligned} \quad \text{(C)}$$

where $a \in A(x^0)$ is some index that is currently active.

In practice, we use an SQP method [32] for the solution of (C) with x^p as the initial point. In the case of convergence issues, one can choose a starting point closer to the Pareto critical point x^0, i.e., $x^p + \lambda(x^0 - x^p)$ for some $\lambda \in (0, 1)$.

3.4 Changing the activation structure

So far, we have seen how the predictor and corrector can be computed when the zero structure is locally constant around x^0. As discussed at the beginning of this section, this allows us to compute the smooth parts of the Pareto critical set in between the kinks. To compute the entire Pareto critical set (or a connected component of it), a mechanism that is able to change the activation structure $A(x)$ is required.

According to Theorem 3.1, an index j can only change from inactive to active (or vice versa) if $x_j = 0$ and $|\nabla f(x)_j| = |\nabla f(x)_a|$ with $a \in A(x)$. We will call such an index j *potentially active* and denote the set of all potentially active indices by

$$\begin{aligned}
A_p(x) = \{j \in \{1, \ldots, n\} \mid &x_j = 0 \text{ and} \\
&|\nabla f(x)_j| = |\nabla f(x)_a|\}.
\end{aligned} \quad (13)$$

3.4.1 Activate indices

Let x^* be a Pareto critical point with $A_p(x^*) \neq \emptyset$. If x^* is an end point of the current activation structure, then the construction of the corrector ensures that we reach it after a finite number of steps (if the step length τ is sufficiently small). If x^* is not an end point, then we will generally not be able to detect it during continuation, as we move with a constant step size. Fortunately, the latter points are less likely to exist, as will be discussed briefly at the end of this section.

Note that not all potentially active indices necessarily need to be activated, such that a way to identify the relevant indices is required. Denoting the number of potentially active indices with p, we have 2^p possible activation structures. For every possible activation structure $A_p^l \subset A_p(x^*)$, $l = 1, \ldots, 2^p$, we can compute a vector $\bar{v}^l = (\bar{v}_1^l, \bar{v}_2^l, \bar{v}_3^l)$ as in the predictor step (Section 3.2). If we assume $|\nabla f(x^*)_a| > 0$, then according to 2(a) in Theorem 3.1, the correct sign of the possibly active indices is given by $-\text{sgn}(\nabla f(x^*))_j$. Therefore, $\bar{v}^l$ is given by

$$\ker \begin{pmatrix} \alpha_1 \nabla^2 f(x^*)_{|A(x^*)\cup A_p^l} & \nabla f(x^*)_{|A(x^*)\cup A_p^l} & s^l \\ 0 & 1 & 1 \end{pmatrix}, \quad (14)$$

with $s^l = -\text{sgn}(\nabla f(x^*))_{|A(x^*)\cup A_p^l}$. If the hessian $\nabla^2 f(x^*)_{|A(x^*)\cup A_p^l}$ is regular, we obtain

$$\begin{aligned}
\bar{v}_1^l &= (1 + |(\nabla f(x^*))_a|^2) \left(\nabla^2 f(x^*)_{|A(x^*)\cup A_p^l}\right)^{-1} s^l, \\
\bar{v}_2^l &= 1, \\
\bar{v}_3^l &= -1.
\end{aligned} \quad (15)$$

Analogously to Section 3.2, we obtain the direction $v_1^l \in \mathbb{R}^n$ by the corresponding embedding $h^1 : \mathbb{R}^{n^A(x^*) + |A_p^l|} \to \mathbb{R}^n$. If A_p^l is a set of potentially active indices that can actually be activated, i.e., if there is a curve γ of Pareto critical points starting in x^* with the active set $A(x^*) \cup A_p^l$, then by construction, v_1^l is the tangent vector of that curve. By analyzing the relationship between the derivative of γ at x^* and the conditions in Theorem 3.1, we obtain necessary conditions for A_p^l to be activated. This is done in the following lemma and the subsequent corollary.

Lemma 3.5. *Let $x^* \in P_c \setminus \{0\}$ with $\nabla f(x^*) \neq 0$. Let $\gamma : (-1, 1) \to \mathbb{R}^n$ such that*

(i) γ is continuously differentiable,
(ii) $\gamma(t) \in P_c \quad \forall t \in [0, 1)$,
(iii) $\gamma(0) = x^*$,
(iv) $A(\gamma(t)) =: A_\gamma$ is constant $\forall t \in (0, 1)$,

Let $a \in A(x^*)$ and $w = \gamma'(0)$. Then for all $j \in A_\gamma \setminus A(x^*)$ we have

(a) either $w_j = 0$ or $\mathrm{sgn}(w_j) = -\mathrm{sgn}((\nabla f(x^*))_j)$,

and for all $j \in A_p(x^*) \setminus A_\gamma$ we get

(b) $\quad \mathrm{sgn}((\nabla f(x^*))_j)(\nabla^2 f(x^*))_j w$
$\quad\quad \leq \mathrm{sgn}((\nabla f(x^*))_a)(\nabla^2 f(x^*))_a w,$

where $(\nabla^2 f(x^*))_j$ is the j-th row of $\nabla^2 f(x^*)$.

Proof. The proof is given in Appendix A. $\square$

Unfortunately, without knowledge of the Pareto critical set, the existence of the curve γ in the previous lemma is almost impossible to verify in practice. Recall that Corollary 3.3 indicates that $\mathcal{P}_c$ is piecewise smooth, where parts in between kinks are embeddings of Pareto critical sets of the smooth problems (5) into $\mathbb{R}^n$. Thus, the existence of $\gamma_{|(0,1)}$ follows from the smooth structure of (5). In light of this, the existence of γ implies a regularity of the smooth parts when they reach a kink. For example, roughly speaking, continuous differentiability of γ on an open neighborhood of $t = 0$ implies that the "curvature" of $\mathcal{P}_c$ is bounded.

Note that the statement of the previous lemma only involves the derivative $w = \gamma'(0)$ and the new active set A_γ. By the construction in (15), w is (up to scaling) equal to the direction v_1^l for the index set $A_p^l = A_\gamma \setminus A(x^*)$. Although A_γ is not known in practice, we can use this relation to identify the indices that need to be activated, as shown in the following corollary.

Corollary 3.6. *Let $x^* \in P_c \setminus \{0\}$ with $\nabla f(x^*) \neq 0$, let γ be a curve as in Lemma 3.5 and let $a \in A(x^*)$.*
Let v_1^l be the vector computed via (15) and the appropriate embedding h^1 corresponding to the index set $A_p^l = A_\gamma \setminus A(x^*)$. Then, for $w = \sigma v_1^l$ with $\sigma \in \{-1, +1\}$,

(a) either $w_j = 0$ or $\mathrm{sgn}(w_j) = -\mathrm{sgn}((\nabla f(x^*))_j)$,

and for all $j \in A_p(x^*) \setminus A_\gamma$,

(b) $\mathrm{sgn}((\nabla f(x^*))_j)(\nabla^2 f(x^*))_j w \leq -\sigma(1 + |(\nabla f(x))_a|)^2$,

where $(\nabla^2 f(x^*))_j$ is the j-th row of $\nabla^2 f(x^*)$.

Remark 3.7. *According to (a) in Corollary 3.6, for all possible activation structures $A_p^l(x^*) \neq \emptyset$, there either exists only one valid direction ($+v_1^l$ or $-v_1^l$) or the tangent vector w of the corresponding curve γ has to satisfy $w_j = 0$ for all $j \in A_p(x^*)$, i.e., γ has to be tangential to a vector with activation structure $A(x^*)$, which is unlikely.*

Using the conditions from Corollary 3.6, we now obtain a predictor for the suitable activation structures with the correct sign. Since we can exclude the direction we are coming from, we have to follow each of the remaining directions. This is shown in Examples 3.8 and 3.9.

Example 3.8. *Solve (MOP-ℓ_1) with*

$$f(x) = (x_1 - 2)^2 + (x_2 - 1)^2 + (x_3 - 1)^2. \qquad (16)$$

Figure 3a shows the Pareto critical set for this problem. Consider $x^* = (1, 0, 0)^\top$. Since $\nabla f(x^*) = (-2, -2, -2)^\top$, we have $A_p(x^*) = \{2, 3\}$ (cf. (13)). According to (15), we get the following possible directions:

$$v_1^0 = \begin{pmatrix} \frac{9}{2} \\ 0 \\ 0 \end{pmatrix}, v_1^1 = \begin{pmatrix} \frac{9}{2} \\ \frac{9}{2} \\ 0 \end{pmatrix}, v_1^2 = \begin{pmatrix} \frac{9}{2} \\ 0 \\ \frac{9}{2} \end{pmatrix}, v_1^3 = \begin{pmatrix} \frac{9}{4} \\ \frac{9}{4} \\ \frac{9}{4} \end{pmatrix}$$

After checking condition (a) in Corollary 3.6, we know that only

$$v_1^0, -v_1^0, v_1^1, v_1^2 \text{ and } v_1^3$$

remain as suitable directions. By checking condition (b), the set of admissible directions can be reduced to

$$-v_1^0 \text{ and } v_1^3.$$

Assuming that we started in the origin, we have to walk in direction v_1^3, i.e., both $j = 2$ and $j = 3$ become active (and positive).

Example 3.9. *Solve (MOP-ℓ_1) with*

$$f(x) = (x_1 - 2)^2 + (x_2 - 1)^2 + 2x_1 x_3. \qquad (17)$$

Figure 3b shows the relevant part of $\mathcal{P}_c$ for this problem. Consider $x^* = (1, 0, 0)^\top$. Since $\nabla f(x^*) = (-2, -2, 2)^\top$, we have $A_p(x^*) = \{2, 3\}$. According to (15), we get as possible directions:

$$v_1^0 = \begin{pmatrix} \frac{9}{2} \\ 0 \\ 0 \end{pmatrix}, v_1^1 = \begin{pmatrix} \frac{9}{2} \\ \frac{9}{2} \\ 0 \end{pmatrix}, v_1^2 = \begin{pmatrix} -\frac{9}{2} \\ 0 \\ 9 \end{pmatrix}, v_1^3 = \begin{pmatrix} -\frac{9}{2} \\ \frac{9}{2} \\ 9 \end{pmatrix}$$

Verification of conditions (a) and (b) in Corollary 3.6 reduces the set of admissible directions to $-v_1^0$, v_1^1 and v_1^2. Assuming that we started in the origin, we have to (try to) advance with the continuation into the directions v_1^1 and $-v_1^2$. A predictor step along v_1^1 directly terminates since the corrector delivers x^* as the nearest Pareto critical point and therefore, the remaining direction is $-v_1^2$. Thus, only $j = 3$ is activated.
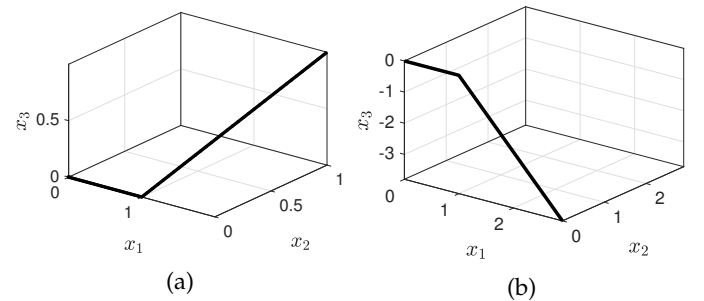

Fig. 3: (a) $\mathcal{P}_c$ for problem (16). (b) Part of $\mathcal{P}_c$ of problem (17).

In practice, often only two possible directions remain after applying Corollary 3.6 – the old and new direction. Nevertheless, it can occur that the Pareto set splits, i.e., that there are at least 3 suitable directions. In this case, the continuation method needs to be advanced in each direction individually. If no direction satisfies the conditions in Corollary 3.6, we have reached the end of a connected component of $\mathcal{P}_c$. In the smooth parts of the Pareto critical set with constant activation structure, a connected component can

only end if $\nabla f(x) = 0$, i.e., a local minimizer for the main objective has been found [33].

Remark 3.10. *Although the requirements of Corollary 3.6 do not hold if $\nabla f(x) = 0$, similar conditions can be obtained for this case. First, note that $\nabla f(x) = 0$ implies that $A_p(x^*)$ is the set of all inactive indices. Since we cannot predict the sign of x_j with $j \in A_p(x^*)$ via the sign of the gradient, we have to consider every possible combination of the signs of potentially active indices for every possible activation structure A_p^l. The conditions of Corollary 3.6 can then be used as necessary conditions, where the sign of the gradient is replaced by the opposite sign of the currently considered activation and sign structure. An additional condition can be obtained by using the fact that $\mathrm{sgn}(\nabla f(x)_j) = -\mathrm{sgn}(x_j)$ for all $j \in A(x)$ has to hold. If the number of inactive indices is large, this results in a very large number of directions that have to be checked. Therefore a more efficient approach needs to be developed in future work for very high-dimensional problems.*

Finally, we note that it is in theory possible that indices are activated in points besides the end points of activation structures. In other words, the Pareto critical set may split into two branches, with one branch continuing along the old structure. Since we move through smooth parts of the Pareto critical set with discrete steps, points like this can not be detected by our method. But based on Corollary 3.6 and Remark 3.7, one can show that such points are unlikely to exist in practice.

3.4.2 Deactivate indices

A situation where indices have to be deactivated is when the predictor crosses zero for some entry, i.e., $\mathrm{sgn}(x_j^p) \neq \mathrm{sgn}(x_j^0)$ for some $j \in A(x^0)$. In this situation, we reduce the predictor step length h to $\bar{h}$ as the largest step size such that

$$x_j^p \leq 0 \text{ if } x_j^0 < 0 \quad \text{and} \quad x_j^p \geq 0 \text{ if } x_j^0 > 0,$$

for all $j \in A(x^0)$, and the indices in $A(x^0) \setminus A(x^p)$ are considered as inactive in the subsequent corrector step. If the corresponding corrector step (C) has a solution, then the continuation method can continue on the new activation structure. Otherwise, we choose a step length in $(0, \bar{h})$ to find a new Pareto critical point with respect to the old active set $A(x^0)$. The likelihood of an active index being falsely deactivated by this mechanism depends on the constant step size τ and the curvature of the Pareto critical set.

A second scenario where active indices can be deactivated is the case where the corrector computes a point x^c where one of the previously active indices is now zero, i.e., $x_i^c = 0$ for some $i \in A(x^0)$. This implies that some of the previously active indices $A(x^0)$ are now only potentially active. In this case, we proceed as in the activation procedure described in Section 3.4.1.

3.5 Implementation & Globalization

We now have all the ingredients to implement Algorithm 1.
Activation of indices:

1) If $A_p(x)$ (Eq. (13)) is non-empty, calculate v^l for all $A_p^l \subset A_p(x)$ using (15),
2) Reduce number of directions using Corollary 3.6,

3) If more than one potential direction remains, proceed in each direction. In most cases, the corrector will yield the information that $\mathcal{P}_c$ only continues in one direction.

Predictor step (when $A_p(x) = \emptyset$):

1) Compute direction $v_1 = h^1(\bar{v}_1)$ using (P),
2) Determine sign of v_1 using (11),
3) Compute the step size along $\pm v_1$ using (12),
4) Determine $x^p = x^0 + h \cdot (\pm v_1)$.

Deactivation of indices: Set the indices j to zero for which $\mathrm{sgn}(x_j^p)$ changes and reduce the predictor step length h to $\bar{h}$ to avoid sign changes.

Corrector step: Compute point $x^c \in \mathcal{P}_c$ using (C) with x^p as the initial condition.

For the starting point, an intuitive choice would be $x_{\text{start}} = 0$, which is Pareto critical and even Pareto optimal as the global minimal point of the ℓ_1-norm. According to Theorem 3.1, the first index that has to be activated in $x_{\text{start}} = 0$ is the one which corresponds to the largest absolute value of the gradient. If there are more than one maximal entries, we can apply the strategy described in Section 3.4. Of course, every other Pareto critical point would also work as a starting point, from which we would have to walk in both directions of the predictor.

If the Pareto critical set consists of multiple connected components, then we have to restart our method once we have reached the end of a connected component. To obtain the required starting points, we can use methods for MOPs like the *reference point* or the ε-*constraint method* [15].

Finally, it is important to note that our method generally computes Pareto *critical* points, and not necessarily Pareto *optimal* points. To obtain the actual Pareto set, an additional non-dominance test is required.

4 NUMERICAL RESULTS

We now study the behavior of the continuation method using three examples. We begin with a nonlinear toy example, and then address the identification of governing equations from data via the SINDy algorithm [1]. Note that the latter is a regression problem and can consequently be adressed by existing homotopy approaches as well as the weighted sum method. Finally, the training of a neural network is studied.

4.1 Toy-Example

We consider (MOP-ℓ_1) with

$$f(x) = \left(x_1 - \frac{1}{4}\right)^2 + \left(x_2 - \frac{1}{2}\right)^2 + (x_3 - 1)^4 - \frac{1}{2}\left(x_3 - \frac{1}{4}\right)^3.$$

Figure 4 shows the result of our continuation method. We observe that indices are activated and deactivated multiple times. Points where the activation structure changes are, $x^1 = (0, 0, 0.375)$, $x^2 = (0, 0.25, \frac{83}{151})$, $x^3 \approx (0, 0.25, 0.81)$, $x^4 \approx (0, 0, 1.07)$, $x^5 \approx (0, 0, 1.93)$ and $x^6 \approx (0, 0.25, 2.014)$. In Figure 4(c) and (d), we see that only the convex part of $\mathcal{P}_c$ set can be computed with the weighted sum method. In particular, a large part where only $j = 3$ is active is missing.
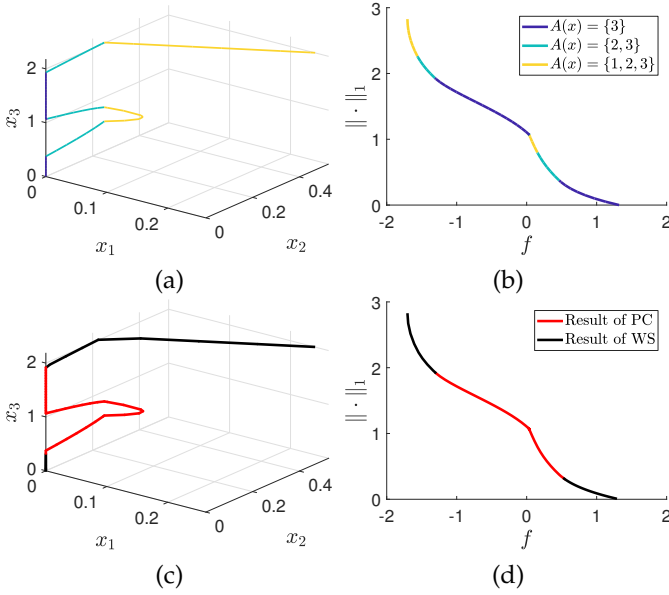
Fig. 4: (a) $\mathcal{P}_c$ for the toy example. (b) Image of $\mathcal{P}_c$. (c) and (d) Comparison to the weighted sum method, with which the red parts cannot be computed.

4.2 SINDy

The second example addresses the sparse identification of nonlinear dynamical systems [1], where the governing system of ordinary (or partial) differential equations is sought:

$$\dot{x}(t) = g(x(t)).$$

Here, $x(t) \in \mathbb{R}^m$ is the m-dimensional system state at time t, whereas the function g describing the system dynamics is unkown. The SINDy approach is then to represent g via a dictionary of c Ansatz functions (denoted by $\Theta(x)$) with the corresponding coefficients $W \in \mathbb{R}^{c \times m}$:

$$g(x(t)) = W^\top \Theta(x(t)).$$

The entries of Θ are arbitrary linear and nonlinear functions such as sine and cosine or polynomials. Using N (potentially noisy) training data points $((x^1, \dot{x}^1), ..., (x^N, \dot{x}^N))$, the objective is to identify a sparse representation of g in terms of Θ, and thus to obtain a robust and interpretable dynamical system from data. The corresponding main objective is thus the following least squares term:

$$f(W) = \frac{1}{mcN} \sum_{i=1}^{N} \left\| W^\top \Theta(x^i) - \dot{x}^i \right\|_2^2.$$

As this is a convex function, the corresponding MOP is also convex and can thus be addressed using the standard penalization approach, i.e., the weighted sum method $\min_W f(W) + \lambda \|W\|_1$ (where W is transformed to one large weight vector). However, as we will see, the solution may be highly sensitive to the choice of λ.

We consider the well-known Lorenz system with parameters $\sigma = 10$, $\rho = 28$ and $\beta = \frac{8}{3}$ for which the dynamical system posses a chaotic solution:

$$\dot{x} = \begin{pmatrix} \sigma \cdot (x_2 - x_1) \\ x_1 \cdot (\rho - x_3) - x_2 \\ x_1 \cdot x_2 - \beta \cdot x_3 \end{pmatrix}.$$

We collect $N = 10,000$ training data points of a single long-term simulation to which we add white noise. The dictionary Θ consists of polynomial terms up to order 3 and is thus comprised of $c = 20$ terms.

The Pareto front of the corresponding MOP – obtained via continuation – is shown in Figure 5 (a), the corresponding non-zero weights and three trajectories are visualized in (b) and (c), respectively. We see that although the Pareto front is convex, choosing the weight in a penalization approach can be challenging. Large parts of the Pareto front are almost parallel to either the horizontal or vertical axis, such that a small change in the penalty weight will result in large jumps along the front (cf. also Figure 1).
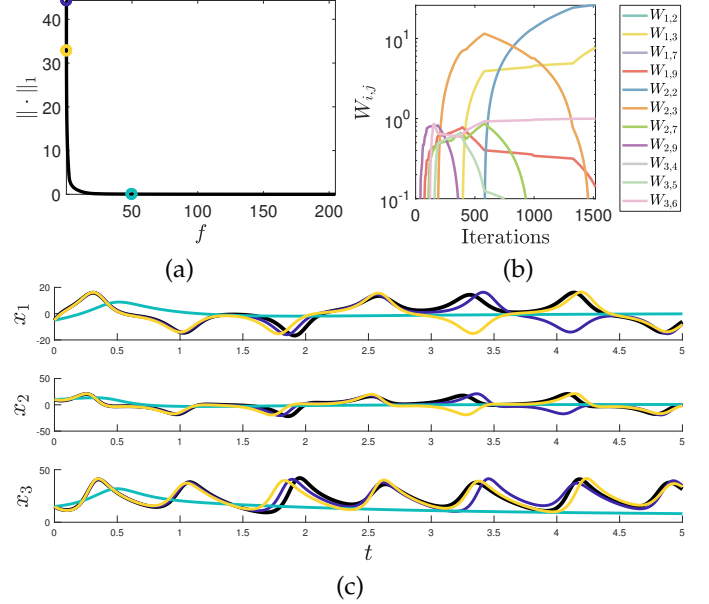

Fig. 5: (a) Convex Pareto front for the SINDy approach applied to the Lorenz system. (b) Evolution of all $W_{i,j}$ which become greater than 0.5 over the iterations. (c) Trajectory of the Lorenz system (black) and the resulting trajectories of the approximated solutions where the corresponding point on the Pareto front is marked in (a).

4.3 Neural Network

Our final example is the training process of the weights W of a feed-forward neural network $y = g_W(x)$ [13]:

$$z_0 = x,$$
$$z_{j+1} = h\left(w_j^\top z_j + w_j^0\right) \quad \text{for } j = 0, \ldots, K-1,$$
$$y = \sigma(z_K),$$

where K is the number of *hidden layers* and σ and h are activation functions. The set of weight matrices, i.e., $W = \{(w_1^0, w_1) \ldots, (w_{K-1}^0, w_{K-1})\}$ (which can again be transformed into a single weight vector), is chosen such that the error on a given training data set $((x^1, y^1), \ldots, (x^N, y^N))$ is minimized. In the case of a classification problem, the error is expressed as the mean of the cross entropy loss, which results in the following objective function:

$$f(W) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{M} (y_c^i \log(g_W(x^i)_c)),$$

where M is the number of classes in the data set and $y_c^i \in \{0, 1\}$ indicates whether x^i belongs to class c. In addition, sparsity is often desired to increase robustness against noisy data.

We here consider a feed forward neural network with $K = 2$ hidden layers. The activation function in the hidden layers is $h(x) = \frac{1}{1+e^{-x}}$, and we chose the softmax function for the output layer, i.e., $\sigma(x)_i = \frac{\exp(x)_i}{\sum_{j=1}^{m}(\exp(x)_j)}$. Each layer consists of two neurons such that the total number of weights is $|W| = 25$. As the training data set, we use the well-known Iris data set [34].
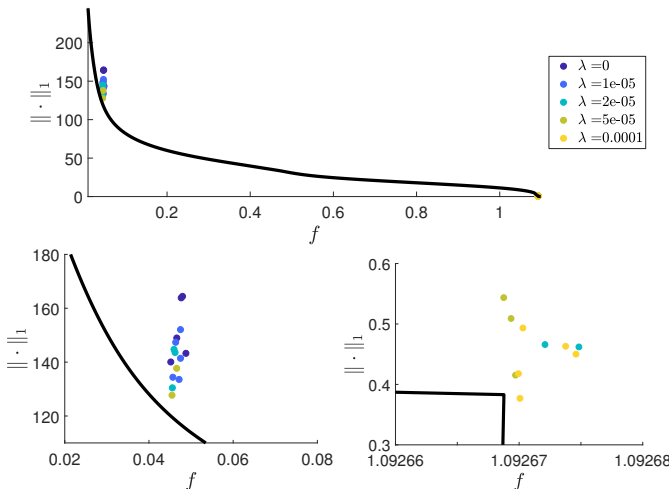

Fig. 6: The Pareto front for the neural network training on the Iris data set is shown in black. The dots show solutions obtained by Adam with different weights for the ℓ_1 penalty term.

Figure 6 shows the Pareto front obtained via the continuation method when starting in $W = 0$. Note that the front is non-convex close to zero. To compare the solution to stochastic gradient descent training, we also show several solutions obtained using the Adam descent method [35] with 10,000 iterations and random initial conditions as well as varying weights λ to penalize $\|W\|_1$. When choosing λ too large, the solution always tends towards zero (which might be due to the non-convexity and the convergence to a local optimum). For small values of λ (including $\lambda = 0$), we obtain points close to the Pareto front. However, these have larger values of $\|W\|_1$ for a comparable training error, and the choice of λ does not yield solutions that differ significantly. This demonstrates the difficulties of the standard penalization approach and the clear advantage of the continuation method, which produces high-quality trade-off solutions right away.

5 Conclusion

The continuation method we present in this paper allows researchers and practitioners with an interest in sparse optimization to systematically calculate the entire set of compromise solutions between sparsity and the main objective, thereby allowing for much more insight as well as informed model selection than weighted-sum approaches or descent methods providing single Pareto optima [35], [36]. The presented approach extends the well-known homotopy methods to the much more complex nonlinear problem setting, where weighting approaches fail due to non-convexity. The optimality conditions specifically derived for this problem class allow us to use continuation methods from smooth multiobjective optimization almost everywhere, and to efficiently analyze and exploit the problem structure in points where non-smoothness occurs.

To unfold the full potential of this multiobjective approach to sparse optimization, several additional questions should be addressed in future work. These contain, for instance, the avoidance of hessian computation (using, e.g., BFGS-like approaches [32]) in order to increase the numerical efficiency, in particular with regard to higher-dimensional problems. This is of particular interest for the training of neural networks with many training parameters. Finally, in order to further extend the range of possible applications, the consideration of additional – more general – non-smooth – objectives as well as constraints (see, for instance, [37]) is certainly of high interest.

References

[1] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 113, no. 15, pp. 3932–3937, 2016.

[2] Y. C. Eldar and G. Kutyniok, *Compressed Sensing: Theory and applications*. Cambridge University Press, 2012.

[3] D. Vidaurre, C. Bielza, and P. Larrañaga, "A survey of L1 regression," *International Statistical Review*, vol. 81, no. 3, pp. 361–387, 2013.

[4] R. G. Baraniuk, E. Candes, M. Elad, and Y. Ma, "Applications of sparse representation and compressive sensing," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 906–909, 2010.

[5] M. Elad, M. A. T. Figueiredo, and Y. Ma, "On the role of sparse and redundant representations in image processing," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 972–982, 2010.

[6] M. D. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M. E. Davies, "Sparse representations in audio and music: From coding to source separation," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 995–1005, 2010.

[7] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[8] S. Chen and D. Donoho, "Basis pursuit," in *Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers*, vol. 1, 1994, pp. 41–44.

[9] M. R. Osborne, B. Presnell, and B. A. Turlach, "A new approach to variable selection in least squares problems," *IMA Journal of Numerical Analysis*, vol. 20, no. 3, pp. 389–403, 2000.

[10] D. M. Malioutov, M. Cetin, and A. S. Willsky, "Homotopy continuation for sparse signal representation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05)*, vol. 5. IEEE, 2005, pp. 733–736.

[11] B. Bringmann, D. Cremers, F. Krahmer, and M. Möller, "The homotopy method revisited: Computing solution paths of ℓ_1-regularized problems," *Mathematics of Computation*, vol. 87, no. 313, pp. 2343–2364, 2018.

[12] D. L. Donoho and Y. Tsaig, "Fast solution of ℓ_1-norm minimization problems when the solution may be sparse," *IEEE Transactions on Information Theory*, vol. 54, no. 11, pp. 4789–4812, 2008.

[13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[14] K. Miettinen, *Nonlinear multiobjective optimization*, 4th ed., ser. International series in operations research & management science. Kluwer Academic Publishing, 2004.

[15] M. Ehrgott, *Multicriteria optimization*, 2nd ed., 2005.

[16] K. Deb, *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, 2001, vol. 16.

[17] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[18] C. Hillermeier, *Nonlinear multiobjective optimization: A generalized homotopy approach*, 2001.

[19] A. Martín and O. Schütze, "Pareto Tracer: a predictor–corrector method for multi-objective optimization problems," *Engineering Optimization*, vol. 50, no. 3, pp. 516–536, 2018.

[20] O. Schütze, O. Cuate, A. Martín, S. Peitz, and M. Dellnitz, "Pareto Explorer: a global/local exploration tool for many-objective optimization problems," *Engineering Optimization*, vol. 52, no. 5, pp. 832–855, 2020.

[21] A. Aşkan and S. Sayın, "SVM classification for imbalanced data sets using a multiobjective optimization framework," *Annals of Operations Research*, vol. 216, no. 1, pp. 191–203, 2014.

[22] H. Aytug and S. Sayın, "Exploring the trade-off between generalization and empirical errors in a one-norm SVM," *European journal of operational research*, vol. 218, no. 3, pp. 667–675, 2012.

[23] H. Li, X. Su, Z. Xu, and Q. Zhang, "MOEA/D with iterative thresholding algorithm for sparse optimization problems," in *Parallel Problem Solving from Nature - PPSN XII*, C. A. Coello Coello, V. Cutello, K. Deb, S. Forrest, G. Nicosia, and M. Pavone, Eds. Springer, 2012, pp. 93–101.

[24] M. Y. Park and T. Hastie, "L1-regularization path algorithm for generalized linear models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 69, no. 4, pp. 659–677, 2007.

[25] S. Rosset, "Tracking curved regularized optimization solution paths," in *Advances in Neural Information Processing Systems*. MIT Press, 2004.

[26] S. Rosset, "Following curved regularized optimization solution paths," in *Advances in Neural Information Processing Systems*, 2005, pp. 1153–1160.

[27] P. Zhao and B. Yu, "Boosted lasso," University of California, Berkeley, Tech. Rep., 2004.

[28] F. H. Clarke, *Optimization and Nonsmooth Analysis*, 1990.

[29] A. Bagirov, N. Karmitsa, and M. M. Mäkelä, *Introduction to Nonsmooth Optimization*. Springer International Publishing, 2014.

[30] M. M. Mäkelä, V. P. Eronen, and N. Karmitsa, "On nonsmooth multiobjective optimality conditions with generalized convexities," in *Optimization in Science and Engineering: In Honor of the 60th Birthday of Panos M. Pardalos*, 2014.

[31] E. L. Allgower and K. Georg, "Numerical continuation methods : An introduction," 1990.

[32] J. Nocedal and S. Wright, *Numerical Optimization*. Springer New York, 2006.

[33] B. Gebken, S. Peitz, and M. Dellnitz, "On the hierarchical structure of Pareto critical sets," *Journal of Global Optimization*, vol. 73, no. 4, p. 891–913, 2019.

[34] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2014.

[36] B. Gebken and S. Peitz, "An efficient descent method for locally Lipschitz multiobjective optimization problems," *Journal of Optimization Theory and Applications (to appear, preprint: arXiv:2004.11578)*, 2020.

[37] J.-C. Loiseau and S. L. Brunton, "Constrained Sparse Galerkin Regression," *Journal of Fluid Mechanics*, vol. 838, pp. 42–67, 2018.

APPENDIX
PROOF OF LEMMA 3.5

First, note that due to continuity of ∇f and γ, we can assume without loss of generality that

$$(\nabla f(\gamma(t)))_j \neq 0 \quad \forall t \in (0,1), \ j \in A_\gamma \cup A_p(x^*). \quad (18)$$

This implies that $\mathrm{sgn}((\nabla f(\gamma(t)))_j)$ is constant for all $t \in [0,1)$ and each $j \in A_\gamma \cup A_p(x^*)$.

(a) Let $j \in A_\gamma \setminus A(x^*)$. From (ii) and (iv) it follows that

$$\mathrm{sgn}(\gamma(t)_j) = -\mathrm{sgn}((\nabla f(\gamma(t)))_j) \quad (19)$$

for all $t \in (0,1)$ and each $j \in A_\gamma$. By the above considerations, this implies that $\mathrm{sgn}(\gamma(t)_j)$ does not change, i.e., is constant on $(0,1)$ for each $j \in A_\gamma$. By (iii) and since $j \notin A(x^*)$, we have $\gamma_j(0) = x_j^* = 0$, so

$$w_j = \gamma_j'(0) = \lim_{t \to 0} \frac{\gamma_j(t) - \gamma_j(0)}{t} = \lim_{t \to 0} \frac{\gamma_j(t)}{t} = \lim_{t \downarrow 0} \frac{\gamma_j(t)}{t}.$$

If $w_j \neq 0$, then (19) implies $\mathrm{sgn}(w_j) = \mathrm{sgn}(\gamma_j(t)) = -\mathrm{sgn}((\nabla f(\gamma(t)))_j)$ for all $t \in (0,1)$. Continuity of $(\nabla f) \circ \gamma$ and $(\nabla f(x^*))_j \neq 0$ (cf. (18)) imply

$$\mathrm{sgn}(w_j) = -\mathrm{sgn}((\nabla f(\gamma(0)))_j) = -\mathrm{sgn}((\nabla f(x^*))_j).$$

(b) Let $j \in A_p(x^*) \setminus A_\gamma$. From (ii) and 2(b) of Theorem 3.1, it follows that

$$\begin{aligned} -\mathrm{sgn}((\nabla f(\gamma(t)))_j)(\nabla f(\gamma(t)))_j &= |(\nabla f(\gamma(t)))_j| \\ \leq |(\nabla f(\gamma(t)))_a| &= -\mathrm{sgn}((\nabla f(\gamma(t)))_a)(\nabla f(\gamma(t)))_a \end{aligned} \quad (20)$$

for all $t \in (0,1)$. As $j \in A_p(x^*)$, we have $|(\nabla f(x^*))_j| = |(\nabla f(x^*))_a|$ and thus, $\mathrm{sgn}((\nabla f(\gamma(0)))_j)(\nabla f(\gamma(0)))_j = \mathrm{sgn}((\nabla f(\gamma(0)))_a)(\nabla f(\gamma(0)))_a$. Hence, from (20) we can conclude

$$\begin{aligned} &\frac{\mathrm{sgn}((\nabla f(\gamma(t)))_j)(\nabla f(\gamma(t)))_j}{t} \\ &\quad - \frac{\mathrm{sgn}((\nabla f(\gamma(0)))_j)(\nabla f(\gamma(0)))_j}{t} \\ &\leq \frac{\mathrm{sgn}((\nabla f(\gamma(t)))_a)(\nabla f(\gamma(t)))_a}{t} \\ &\quad - \frac{\mathrm{sgn}((\nabla f(\gamma(0)))_a)(\nabla f(\gamma(0)))_a}{t} \end{aligned} \quad (21)$$

for all $t \in (0,1)$. Recall that $\mathrm{sgn}((\nabla f(\gamma(t)))_j)$ and $\mathrm{sgn}((\nabla f(\gamma(t)))_a)$ are constant with respect to $t \in [0,1)$. Thus, (21) is equivalent to

$$\begin{aligned} &\mathrm{sgn}((\nabla f(\gamma(0)))_j)\frac{(\nabla f(\gamma(t)))_j - (\nabla f(\gamma(0)))_j}{t} \\ &\leq \mathrm{sgn}((\nabla f(\gamma(0)))_a)\frac{(\nabla f(\gamma(t)))_a - (\nabla f(\gamma(0)))_a}{t} \end{aligned}$$

for all $t \in (0,1)$. Thus, taking the limit $t \downarrow 0$ on both sides of (21) yields

$$\begin{aligned} &\mathrm{sgn}((\nabla f(\gamma(0)))_j)(\nabla^2 f(\gamma(0)))_j \gamma'(0) \\ &\leq \mathrm{sgn}((\nabla f(\gamma(0)))_a)(\nabla^2 f(\gamma(0)))_a \gamma'(0), \end{aligned}$$

where $(\nabla^2 f(\gamma(0)))_j$ is the j-th row of $\nabla^2 f(\gamma(0))$. Replacing $\gamma(0) = x^*$ and $\gamma'(0) = w$ yields

$$\begin{aligned} &\mathrm{sgn}((\nabla f(x^*))_j)(\nabla^2 f(x^*))_j w \\ &\leq \mathrm{sgn}((\nabla f(x^*))_a)(\nabla^2 f(x^*))_a w. \end{aligned}$$