

Gebken, Bennet; Peitz, Sebastian

Article — Published Version

Inverse multiobjective optimization: Inferring decision criteria from data

Journal of Global Optimization

Provided in Cooperation with:

Springer Nature

Suggested Citation: Gebken, Bennet; Peitz, Sebastian (2020) : Inverse multiobjective optimization: Inferring decision criteria from data, Journal of Global Optimization, ISSN 1573-2916, Springer US, New York, NY, Vol. 80, Iss. 1, pp. 3-29, <https://doi.org/10.1007/s10898-020-00983-z>

This Version is available at:

<https://hdl.handle.net/10419/288621>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



<https://creativecommons.org/licenses/by/4.0/>



Inverse multiobjective optimization: Inferring decision criteria from data

Bennet Gebken¹  · Sebastian Peitz¹

Received: 28 November 2019 / Accepted: 11 December 2020 / Published online: 26 December 2020
© The Author(s) 2020

Abstract

It is a challenging task to identify the objectives on which a certain decision was based, in particular if several, potentially conflicting criteria are equally important and a continuous set of optimal compromise decisions exists. This task can be understood as the inverse problem of multiobjective optimization, where the goal is to find the objective function vector of a given Pareto set. To this end, we present a method to construct the objective function vector of an unconstrained multiobjective optimization problem (MOP) such that the Pareto critical set contains a given set of data points with prescribed KKT multipliers. If such an MOP can not be found, then the method instead produces an MOP whose Pareto critical set is at least close to the data points. The key idea is to consider the objective function vector in the multiobjective KKT conditions as variable and then search for the objectives that minimize the Euclidean norm of the resulting system of equations. By expressing the objectives in a finite-dimensional basis, we transform this problem into a homogeneous, linear system of equations that can be solved efficiently. Potential applications of this approach include the identification of objectives (both from clean and noisy data) and the construction of surrogate models for expensive MOPs.

Keywords Multiobjective optimization · Inverse optimization · Pareto set · Pareto critical set

Mathematics Subject Classification 49N45 · 90C29

1 Introduction

When applying optimization to real-world problems, there are often multiple quantities that have to be optimized at the same time. In production for example, typical goals are the maximization of the quality of a product and the minimization of the production cost. When

This research was funded by the DFG Priority Programme 1962 “Non-smooth and Complementarity-based Distributed Parameter Systems”.

✉ Bennet Gebken
bgebken@math.upb.de

¹ Chair of Applied Mathematics, Paderborn University, Warburger Str. 100, 33098 Paderborn, Germany

the objectives are conflicting, there cannot be a single solution that is optimal for all objectives at the same time. This is called a *multiobjective optimization problem (MOP)*. To solve a problem like this, we search for the set of all optimal compromises, the so-called *Pareto set*, containing all *Pareto optimal points*. A point x^* is called Pareto optimal if there exists no other point that is at least as good as x^* in all objectives, but strictly better than x^* in at least one objective.

While most of the research in multiobjective optimization is concerned with efficiently computing the Pareto set of a given MOP, we here address the *inverse problem of multiobjective optimization*:

Given a set $P \subseteq \mathbb{R}^n$, identify the objectives for which P
is the Pareto set. (IMOP)

Although it is possible to state this problem in such a general form, it will have many degenerate solutions that we are not interested in, since there is no restriction on any type of regularity of the objective functions. Therefore, we will instead consider a more well-behaved version of this problem that arises by using the concept of *Pareto criticality*. A point $x^* \in \mathbb{R}^n$ is called *Pareto critical* if it satisfies the *Karush–Kuhn–Tucker (KKT) conditions* [24], i.e., if there is a convex combination of all the gradients of the objective functions $f_i \in C^1(\mathbb{R}^n, \mathbb{R})$, $i \in \{1, \dots, k\}$, in x^* which is zero. In that case, if $\alpha^* \in \mathbb{R}^k$ contains the coefficients of this convex combination, then α^* is called a *KKT vector of x^** and the pair (x^*, α^*) is called an *extended Pareto critical point*. The set of all such pairs is called the *extended Pareto critical set*. The above problem can be restated using this concept:

Given a finite data set $\mathcal{D} = (\mathcal{D}_x, \mathcal{D}_\alpha) \subseteq \mathbb{R}^n \times \mathbb{R}^k$, find an obj. fun.
vector $f \in C^1(\mathbb{R}^n, \mathbb{R}^k)$ whose extended Pareto critical set contains \mathcal{D} . (IMOPc)

Since the search space $C^1(\mathbb{R}^n, \mathbb{R}^k)$ is infinite-dimensional, we will consider finite-dimensional subspaces of $C^1(\mathbb{R}^n, \mathbb{R})$ that are spanned by sets of basis functions $\mathcal{B} \subseteq C^1(\mathbb{R}^n, \mathbb{R})$. This will transform (IMOPc) into a homogeneous linear system in the coefficients of the basis functions which can be solved by singular value decomposition. In practice, an exact solution of (IMOPc) is unlikely to exist, due to noise in the data and the finite-dimensional approximation of $C^1(\mathbb{R}^n, \mathbb{R}^k)$. Thus, we will also present an algorithm to generate objective function vectors whose Pareto critical sets (and corresponding KKT vectors) are close to the data set.

Part of the reason why (IMOPc) is more well-behaved than the original (IMOP) is the fact that we assume the KKT vectors to be given in the data. Geometrically, if we attach the KKT vector of a Pareto optimal point x^* to $f(x^*)$, then it is orthogonal to the Pareto front (cf. [20], Theorem 4.2). Thus, the assumption that the KKT vectors are available means that the data must provide significantly more information than just the Pareto critical points. Nevertheless, we will later show that there are applications where this data is available (or can be obtained). The first application is the generation of test problems for MOP solvers, where the choice of KKT vectors can be made by considering geometrical and topological results of Pareto critical sets (cf. [17,20,26]). The second application is the construction of surrogate models for expensive MOPs, where the idea is to compute a few optimal points of the expensive problem and to then use these points as data for our inverse approach. Most standard methods for solving the expensive MOP will also (either explicitly or implicitly) provide the corresponding KKT vectors that are needed. For example, in the weighting method (cf. [27]), the KKT vector of an optimal point is the weighting vector that was used to compute it. We point out that while our approach works well in most of the examples we

present in this paper, there remain challenges that need to be addressed to increase the range of applicability, in particular for the generation of surrogate models. These open problems will be discussed in the conclusion.

For the single objective case, i.e., for $k = 1$, problems like (IMOP) are addressed within the field of *inverse optimization*. For combinatorial problems, a survey on inverse optimization can be found in [19]. In [21] and [1], inverse linear problems of the form $\min_x c^\top x$ (with some linear constraints) were considered, where the goal is to find the cost vector c so that a given feasible point is optimal. In [23], convex parameter-dependent problems were considered with the intention of estimating the objective function from observations of parameter values and associated optimal solutions. Similarly to our approach in this paper, this is done by expressing the objective function as a linear combination of some pre-selected basis functions and then minimizing the residuals of the first-order optimality conditions in the given observations. Part of the literature in the single objective case is concerned with finding a weighting vector for the objectives of an MOP such that a given feasible point is Pareto optimal (cf. [6,7]). This area is also referred to as *inverse multiobjective optimization*, but differs from the context in this paper. Recently, a first result in the multiobjective case has appeared. In [12], a method was presented to find the parameters of a parameter-dependent, convex and constrained MOP such that its Pareto set contains a set of given, noisy points (modeled via probability distributions). This was done by discretizing the Pareto set for a fixed parameter by a finite number of solutions of the weighting method and then minimizing the sum over the distances of the given points to the discretization. This strategy was formalized as a mixed integer linear problem, for which a heuristic solution method was proposed.

By interpreting the coefficients of the linear combination of basis functions in our approach as parameters of a parameter-dependent MOP, we are in a similar situation as in [12]. In contrast to the approach in [12], we will not require convexity of the objective functions and not rely on heuristic methods, but only consider the unconstrained case where the KKT vectors are known. On the one hand, this will make our approach more restrained, since the assumption of knowing the KKT vectors is strong, as discussed above. On the other hand, if the KKT vectors are known (as in the applications presented in this paper), allowing non-convex objectives will significantly increase the complexity of the geometry and topology of the data that can be handled. This will also be highlighted in our examples, where many of the objective functions are non-convex. Furthermore, the approach in [12] to avoid the assumption of knowing KKT vectors (via the weighting method) strongly depends on the convexity of the objective functions and thus can not be used in the non-convex setting.

The remainder of this article is structured as follows. We begin with a brief introduction to multiobjective optimization in Sect. 2 before presenting our main theoretical results in Sect. 3. There, we will first investigate the existence of an objective function vector in the span of the chosen basis \mathcal{B} for which the data points are extended Pareto critical. We then address the task of finding the objective function vector whose extended Pareto critical set is as close to a given data set as possible. Afterwards, we apply our algorithm to three fundamentally different problem classes:

- the construction of objective functions from prescribed data (Sect. 4.1),
- the identification of objective functions of stochastic MOPs (Sect. 4.2),
- the generation of surrogate problems in situations where the objective functions are known, but expensive to evaluate (Sect. 5).

In particular, we will discuss how the KKT vectors for the data can be obtained for each problem class. Finally, we draw a conclusion and discuss open problems of our method in Sect. 6.

For our numerical results, we use the built-in method `svd` from MATLAB 2017a for singular value decomposition. For the computation of the extended Pareto critical sets in this article, we use the Continuation Method `CONT-RECOVER` from [34].

2 Multiobjective optimization

In this section, we will briefly introduce the basic concepts of multiobjective optimization. For a more detailed introduction, we refer to [14,20,27].

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}^k$ be a vector-valued function, called the *objective function vector*, with continuously differentiable components $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i \in \{1, \dots, k\}$, called *objective functions*. It maps the *variable space* \mathbb{R}^n to the *image space* \mathbb{R}^k . The goal of multiobjective optimization is to minimize the objective function vector f , i.e., to minimize all objective functions f_i simultaneously. This is called a *multiobjective optimization problem (MOP)* and is denoted by

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

In contrast to scalar optimization (i.e., $k = 1$), it is not immediately clear what we mean by minimizing f , as there is no natural total order of the objective values in \mathbb{R}^k for $k > 1$. As a result, we cannot expect to find a single point that solves (MOP). Instead, we search for the *Pareto set* which is defined in the following way:

- Definition 1** (a) A point $x^* \in \mathbb{R}^n$ *dominates* a point $x \in \mathbb{R}^n$, if $f_i(x^*) \leq f_i(x)$ for all $i \in \{1, \dots, k\}$ and $f_j(x^*) < f_j(x)$ for some $j \in \{1, \dots, k\}$.
- (b) A point $x^* \in \mathbb{R}^n$ is called *locally Pareto optimal* if there exists an open set $U \subseteq \mathbb{R}^n$ with $x^* \in U$ such that there is no point $x \in U$ dominating x^* . If this holds for $U = \mathbb{R}^n$, then x^* is called *Pareto optimal*.
- (c) The set of all (locally) Pareto optimal points is called the *(local) Pareto set*, its image under f the *(local) Pareto front*.

Similar to scalar optimization, there are necessary conditions for local Pareto optimality using the first-order derivatives of f , called the *Karush–Kuhn–Tucker (KKT) conditions* [20]:

Theorem 1 Let x^* be a locally Pareto optimal point of (MOP) and

$$\Delta_k := \left\{ \alpha \in (\mathbb{R}^{\geq 0})^k : \sum_{i=1}^k \alpha_i = 1 \right\}. \quad (1)$$

Then there exists some $\alpha^* \in \Delta_k$ such that

$$Df(x^*)^\top \alpha^* = \sum_{i=1}^k \alpha_i^* \nabla f_i(x^*) = 0. \quad (\text{KKT})$$

For $k = 1$, (KKT) reduces to the well-known optimality condition $\nabla f(x^*) = 0$. The set of points satisfying the KKT conditions is a superset of the (local) Pareto set and we make the following definition:

Definition 2 Let $x \in \mathbb{R}^n$.

- (a) If there exists some $\alpha \in \Delta_k$ (with Δ_k as in (1)) such that (KKT) holds, then x is called *Pareto critical* and α a *KKT vector* of x containing the *KKT multipliers* $\alpha_i, i \in \{1, \dots, k\}$. The set of Pareto critical points P_c of (MOP) is called the *Pareto critical set*. (Pareto critical points are sometimes also referred to as *stationary points* by other authors.)
- (b) In the situation of a), the pair $(x, \alpha) \in \mathbb{R}^n \times \Delta_k$ is called an *extended Pareto critical point*. The set of all such pairs $P_M \subseteq \mathbb{R}^n \times \Delta_k$ is called the *extended Pareto critical set*.

Since the structure of P_c and P_M will be important for our approach, we will briefly mention three results on this topic: In [9,26] it was shown that P_c is generically a stratification, which basically means that it is a “manifold with boundaries and corners”. In [17] it was shown that the boundary (or *edge*) of P_c is covered by Pareto critical sets of subproblems where only subsets of the set of objective functions are optimized. In [21] it was shown that $\{(x, \alpha) \in P_M : \alpha \in (\mathbb{R}^{>0})^k\} \subseteq P_M$ is a $(k - 1)$ -dimensional submanifold of \mathbb{R}^{n+k} if a certain rank condition holds.

3 Inferring objective function vectors from data

We will now present a way to construct objective function vectors for which P_M contains a finite set of given data points $\mathcal{D}_x = \{\bar{x}^1, \dots, \bar{x}^N\} \subseteq \mathbb{R}^n$ with corresponding KKT vectors $\mathcal{D}_\alpha = \{\bar{\alpha}^1, \dots, \bar{\alpha}^N\} \subseteq \Delta_k$. The general concept of this inverse approach is to consider x^* and α^* as given in (KKT) instead of the objective function vector f . So in contrast to the usual task of searching for an $x \in \mathbb{R}^n$ for which an $\alpha \in \Delta_k$ exists so that (KKT) holds, we now search for an $f \in C^1(\mathbb{R}^n, \mathbb{R}^k)$ for which (KKT) holds for all \bar{x}^j and $\bar{\alpha}^j, j \in \{1, \dots, N\}$. As it is infinite-dimensional, we obviously cannot search the entire $C^1(\mathbb{R}^n, \mathbb{R}^k)$. Instead, we consider finite-dimensional linear subspaces of $C^1(\mathbb{R}^n, \mathbb{R})$ that are spanned by a set of basis functions $\mathcal{B} = \{b_1, \dots, b_d\} \subseteq C^1(\mathbb{R}^n, \mathbb{R})$, and then search for $f \in \text{span}(\mathcal{B})^k$. An example for the choice of basis functions is the monomials in n variables such that $\text{span}(\mathcal{B})$ is the space of polynomials (up to a certain degree). The usage of basis functions reduces the task of finding an $f \in C^1(\mathbb{R}^n, \mathbb{R}^k)$ to the task of finding the coefficients $c \in \mathbb{R}^d$ of the corresponding linear combination of basis functions. This problem can be stated as a homogeneous linear problem in c and can be solved efficiently via singular value decomposition. Furthermore, if no objective function vector exists whose extended Pareto critical set exactly contains the data, e.g., due to noise or a bad choice of basis functions, then approximate solutions of this linear problem yield objective function vectors whose extended Pareto critical set is at least close to the data. In particular, the smallest singular value can be used as a measure of how well the given data set can be represented as an extended Pareto critical set of a function consisting of the given basis functions.

We will assume for the remainder of this section that the following are given:

- a data set $\mathcal{D} = \{(\bar{x}^1, \bar{\alpha}^1), (\bar{x}^2, \bar{\alpha}^2), \dots, (\bar{x}^N, \bar{\alpha}^N)\} \subseteq \mathbb{R}^n \times \Delta_k$ (and in particular the number of objective functions k),
- a set of basis functions $\mathcal{B} = \{b_1, \dots, b_d\} \subseteq C^1(\mathbb{R}^n, \mathbb{R})$ with linearly independent derivatives.

As discussed in the introduction, the assumption that the KKT vectors are given in the data is relatively strong. Nevertheless, in Sects. 4 and 5 we will present applications where this data can be obtained. Depending on the application, this is done in two different ways:

- At the end of Sect. 2, we summarized some of the structural results about the extended Pareto critical set P_M . Since we want to generate an MOP such that the data is contained

in $P_{\mathcal{M}}$, these results also impact our data set \mathcal{D} , and in particular the KKT vectors. While the structural results are clearly not strong enough to uniquely determine the data \mathcal{D}_α for the KKT vectors from the data \mathcal{D}_x of the Pareto critical points, they can still be useful. This is mainly the case when we want to use our approach to generate test problems for MOP solvers (Sect. 4.1), where we can influence topological properties like the boundary and the connectedness of P_c via the KKT vectors. For example, if \bar{x}^j should lie on the boundary of P_c , then one component of the corresponding KKT vector $\bar{\alpha}^j$ has to be zero.

- In the forward problem of computing the Pareto set of a known objective function vector, the KKT vectors can often be obtained as a by-product of the solution method. For example, for many scalarization techniques like the weighting method, the KKT vectors can be derived from the first-order optimality conditions of the scalar problem (see Sect. 5). Thus, in the context of generating surrogate models for MOPs, we can obtain the data for the KKT vectors by computing a few Pareto optimal points of the original problem via one of these techniques. This is demonstrated in Sect. 4.2 (for stochastic MOPs) and Sect. 5 (for computationally expensive MOPs).

3.1 Existence of exact approximations

In this subsection, our goal is to find an objective function vector with components in $\text{span}(\mathcal{B})$ for which the set \mathcal{D} is exactly extended Pareto critical. In other words, our goal is to find a function $f : \mathbb{R}^n \rightarrow \mathbb{R}^k$, $f = (f_i)_{i \in \{1, \dots, k\}}$, $f \neq 0$ with $f_i \in \text{span}(\mathcal{B}) \forall i \in \{1, \dots, k\}$ and

$$Df(\bar{x})^\top \bar{\alpha} = 0 \quad \forall (\bar{x}, \bar{\alpha}) \in \mathcal{D}. \tag{2}$$

To this end, for $f_i \in \text{span}(\mathcal{B})$, we can write

$$f_i(x) = \sum_{j=1}^d c_{ij} b_j(x) \tag{3}$$

for some $c_i \in \mathbb{R}^d$. Thus, we obtain

$$\begin{aligned} Df(x)^\top \alpha &= \sum_{i=1}^k \alpha_i \nabla f_i(x) = \sum_{i=1}^k \alpha_i \sum_{j=1}^d c_{ij} \nabla b_j(x) = \sum_{i=1}^k \sum_{j=1}^d \alpha_i c_{ij} \nabla b_j(x) \\ &= L(x, \alpha) c \end{aligned}$$

with

$$c := (c_{11}, \dots, c_{1d}, c_{21}, \dots, c_{2d}, \dots, c_{k1}, \dots, c_{kd})^\top \in \mathbb{R}^{k \cdot d} \tag{4}$$

and

$$\begin{aligned} L(x, \alpha) &:= (\alpha_1 \nabla b_1(x), \dots, \alpha_1 \nabla b_d(x), \alpha_2 \nabla b_1(x), \dots, \alpha_2 \nabla b_d(x), \dots, \alpha_k \nabla b_1(x), \dots, \alpha_k \nabla b_d(x)) \\ &\in \mathbb{R}^{n \times (k \cdot d)}. \end{aligned}$$

Let

$$\mathcal{L} := \begin{pmatrix} L(\bar{x}^1, \bar{\alpha}^1) \\ \vdots \\ L(\bar{x}^N, \bar{\alpha}^N) \end{pmatrix} \in \mathbb{R}^{(n \cdot N) \times (k \cdot d)}.$$

Then (2) is equivalent to the homogeneous linear system

$$\mathcal{L}c = 0. \tag{5}$$

Since the derivatives of the basis functions are linearly independent, a (nontrivial) function satisfying (2) exists if and only if

$$rk(\mathcal{L}) < k \cdot d. \tag{6}$$

We will now consider two cases for the dimension of system (5):

Case 1: (5) is an underdetermined system, i.e., $n \cdot N < k \cdot d$. In this case, (6) automatically holds such that (5) possesses at least one nontrivial solution. In other words, $dim(ker(\mathcal{L})) > 0$. Note that in this case, our approach resembles an interpolation method. In fact, for $n = 1$, $k = 1$ and monomial basis functions, \mathcal{L} is similar to the *Vandermonde matrix* from polynomial interpolation (without the constant column).

Case 2: (5) is a square or overdetermined system, i.e., $n \cdot N \geq k \cdot d$. This means that generically, (5) does not have a solution, and we have to check the condition (6). In practice, we can use singular value decomposition (SVD) to do this, as the rank of \mathcal{L} equals the number of singular values of \mathcal{L} that are non-zero. In particular, as $rk(\mathcal{L}) = k \cdot d - dim(ker(\mathcal{L}))$, it yields the dimension of the solution space of (5).

For ease of notation, we make the following definition:

Definition 3 Let

$$\mathcal{F} : \mathbb{R}^{k \cdot d} \rightarrow C^1(\mathbb{R}^n, \mathbb{R}^k), \quad c \mapsto (f_i)_{i \in \{1, \dots, k\}} = \left(\sum_{j=1}^d c_{ij} b_j \right)_{i \in \{1, \dots, k\}}$$

be the map that maps a coefficient vector c onto the corresponding objective function vector $(f_i)_{i \in \{1, \dots, k\}}$ (cf. (3) and (4)).

It is easy to see that \mathcal{F} is linear and by the linear independence of the derivatives of the basis functions, \mathcal{F} is also injective.

3.2 Finding the best approximation

In most applications, one can expect that (5) is overdetermined and that it cannot be solved exactly. Even if there was a solution, we would require exact data to find it, which is numerically impossible. Furthermore, the case where the data is slightly noisy is much more realistic. Therefore, it makes more sense to look for the MOP whose extended Pareto critical set is the best approximation for a given data set, i.e., where $\|Df(\bar{x})^\top \bar{\alpha}\|_2$ is as small as possible for all $(\bar{x}, \bar{\alpha}) \in \mathcal{D}$. To this end, consider the problem

$$\min_{\|c\|_2=1} \|\mathcal{L}c\|_2, \tag{7}$$

where the vector of coefficients is constrained to the unit sphere $S^{(k \cdot d)-1}$ in $\mathbb{R}^{k \cdot d}$ to avoid the trivial solution $c^* = 0$. If c^* is a solution of (7) and $f = \mathcal{F}(c^*)$ is the corresponding objective function vector, then

$$\|Df(\bar{x})^\top \bar{\alpha}\|_2 = \|L(\bar{x}, \bar{\alpha})c^*\|_2 \leq \|\mathcal{L}c^*\|_2 \quad \forall (\bar{x}, \bar{\alpha}) \in \mathcal{D}, \tag{8}$$

i.e., the optimal value of (7) is an upper bound for all $\|Df(\bar{x})^\top \bar{\alpha}\|_2$ with $(\bar{x}, \bar{\alpha}) \in \mathcal{D}$. In particular, the optimal value of (7) is zero if and only if (6) holds. Problem (7) can easily be

solved using SVD (see, e.g., [16]): Assume that $n \cdot N \geq k \cdot d$, i.e., (5) is overdetermined. Let

$$\mathcal{L} = USV^\top$$

be the SVD of \mathcal{L} with sorted singular values $s_1 \leq s_2 \leq \dots \leq s_{k \cdot d}$. Let $v_1, \dots, v_{k \cdot d} \in \mathbb{R}^{k \cdot d}$ be the right-singular vectors of \mathcal{L} , i.e., the columns of V . Then

$$\min_{\|c\|_2=1} \|\mathcal{L}c\|_2 = s_1 \quad \text{and} \quad \arg \min_{\|c\|_2=1} \|\mathcal{L}c\|_2 = \text{span}(\{v_i : s_i = s_1\}) \cap \mathcal{S}^{(k \cdot d)-1}. \quad (9)$$

Consequently, s_1 is a measure for how well the data set \mathcal{D} can be approximated with the extended Pareto critical set of an MOP where the objective functions are linear combinations of the basis functions in \mathcal{B} . Furthermore, the singular values of \mathcal{L} can be used to determine the dimension of the space of approximating objective function vectors.

Algorithm 1 Generate objective function vector from data

Given: Data set $\mathcal{D} \subseteq \mathbb{R}^n \times \Delta_k$, basis functions $\mathcal{B} \subseteq C^1(\mathbb{R}^n, \mathbb{R})$, threshold \bar{s} .

- 1: Assemble \mathcal{L} .
- 2: Calculate the SVD of \mathcal{L} with singular values $s_1 \leq s_2 \leq \dots \leq s_{k \cdot d}$ and right-singular vectors $v_1, \dots, v_{k \cdot d}$.
- 3: Identify the indices $I = \{1, \dots, i^*\}$, $i^* \leq k \cdot d$, such that $s_i \leq \bar{s}$ for all $i \in I$.
- 4: Choose an element

$$c^* \in \text{span}(\{v_i : i \in I\}) \setminus \{0\} \subseteq \mathbb{R}^{k \cdot d}$$

with $\|c^*\|_2 = 1$.

- 5: Assemble the objective function vector $f = \mathcal{F}(c^*)$ as in (3).
-

Algorithm 1 summarizes the numerical procedure which follows from the above considerations. The resulting approximation then satisfies the following property:

Theorem 2 *Let f be the result of Algorithm 1 and s_{i^*} be the largest singular value less or equal to \bar{s} . Then*

$$\|Df(\bar{x})^\top \bar{\alpha}\|_2 \leq s_{i^*} \quad \forall (\bar{x}, \bar{\alpha}) \in \mathcal{D}.$$

In particular, if $s_{i^} = 0$, then all $(\bar{x}, \bar{\alpha}) \in \mathcal{D}$ are extended Pareto critical for the MOP with objective function vector f .*

Proof Let c^* be the coefficient vector in step 4 such that $f = \mathcal{F}(c^*)$. Then there is some $\lambda \in \mathbb{R}^{k \cdot d}$ with $c^* = V\lambda$, $\lambda_{i^*+1} = \dots = \lambda_{k \cdot d} = 0$ and $1 = \|c^*\|_2 = \|V\lambda\|_2 = \|\lambda\|_2$. Thus

$$\begin{aligned} \|\mathcal{L}c^*\|_2 &= \|\mathcal{L}V\lambda\|_2 = \|US\lambda\|_2 = \|S\lambda\|_2 = \sqrt{\sum_{i=1}^{k \cdot d} s_i^2 \lambda_i^2} \leq s_{i^*} \sqrt{\sum_{i=1}^{k \cdot d} \lambda_i^2} \\ &= s_{i^*} \|\lambda\|_2 = s_{i^*}. \end{aligned}$$

Combining this with (8) completes the proof. □

Some properties of Algorithm 1 are highlighted in the following remark.

Remark 1 (a) Algorithm 1 can also be applied when (5) is underdetermined, i.e., when $n \cdot N \leq k \cdot d$, by treating $v_{(n \cdot N)+1}, \dots, v_{k \cdot d}$ as right-singular vectors to the ‘‘singular value’’ zero.

(b) In general, if $i^* > 1$, there is no obvious choice for c in step 4. A possible approach is to choose c as sparse as possible (using, e.g., L_1 minimization [35]). This can be very advantageous for interpretability, see also [5] for sparse identification in the dynamical systems context.

- (c) It is important to note that by construction, if $s_{i^*} = 0$, we can only guarantee that \mathcal{D} is a subset of the extended Pareto critical set of f . It is possible for the extended Pareto critical set to contain more than just \mathcal{D} (cf. Example 1 and 3). Therefore, there are cases where the smallest singular value is 0, but the corresponding MOP might not be desirable.
- (d) According to (9), if $s_{i^*} = 0$, we can take any element of $\text{span}(\{v_i : i \in I\}) \setminus \{0\}$ in step 3 and do not need to normalize it.

We will conclude this section with a brief discussion on the choice of the set of basis functions \mathcal{B} . It should satisfy the following requirements:

- (i) The derivatives of the basis functions should be linearly independent to avoid trivial solutions. (In particular, this implies that the representation of the derivatives of elements of $\text{span}(\mathcal{B})$ via coefficients of the derivatives of elements of \mathcal{B} is unique.)
- (ii) Since we have to evaluate the derivatives of the basis functions in every data point in \mathcal{D}_x for the assembly of \mathcal{L} , the evaluation of these derivatives should be efficient.
- (iii) In practice, an initial, a priori choice of \mathcal{B} will often be insufficient. Thus, it should be possible to increase the quality of the approximation by increasing the size of \mathcal{B} without much effort.

An intuitive choice for \mathcal{B} are the monomials in n variables up to degree $l \in \mathbb{N}$, i.e.,

$$\mathcal{B} = \{x_1^{l_1} x_2^{l_2} \cdots x_n^{l_n} : l_i \in \mathbb{N} \cup \{0\}, i \in \{1, \dots, n\}, 0 < l_1 + l_2 + \dots + l_n \leq l\}.$$

It is easy to see that (i) and (ii) are satisfied for this choice. For (iii), the Stone-Weierstrass theorem (cf. [31]) implies that for any compact $D \subseteq \mathbb{R}^n$ and any $g \in C^1(D, \mathbb{R})$, there exists a sequence of polynomials on D that converges to g (with respect to $\|\cdot\|_\infty$). Thus, for $g \in C^1(\mathbb{R}^n, \mathbb{R}^k)$, uniform convergence with polynomial functions can be guaranteed component-wise on compact subsets of \mathbb{R}^n . Therefore, for the rest of this article, we will always consider the monomials up to a fixed degree as the set of basis functions.

4 Application 1: Constructing objectives from clean and noisy data

In this section, we will show how the results from Sect. 3 can be utilized to construct objective functions from clean and noisy data. Our first scenario will be the construction of test problems for MOP solvers, where the data comes from a discretization of some continuous (i.e., non-discrete) set. In the second scenario, we will consider stochastic MOPs, where we will reconstruct the expected value of the objective function vectors using stochastic (i.e., noisy) solution data.

4.1 Inferring objectives from exact data

Test problems and generators of test problems are an important tool to investigate the behavior and to benchmark MOP solvers (cf. [10,22,37]). The idea is to interpret our method from Sect. 3 as a way to generate MOPs with a prescribed extended Pareto critical set. Thus, instead of finitely many extended Pareto critical points, the goal here is to prescribe the complete set. We do this by (formally) constructing an infinite (or “continuous”) data set $\mathcal{D}^\infty = (\mathcal{D}_x^\infty, \mathcal{D}_\alpha^\infty) \subseteq \mathbb{R}^n \times \Delta_k$, and then using an even discretization of \mathcal{D}^∞ with $N \in \mathbb{N}$ points (for large N) as the input for Algorithm 1. If the smallest singular value of \mathcal{L} is zero, \mathcal{D}^∞ will at least be contained in the extended Pareto critical set of the resulting MOP.

In this application, the KKT vectors in $\mathcal{D}_\alpha^\infty$ have to be chosen such that \mathcal{D}^∞ has the topological and geometrical properties of an extended Pareto critical set. This is nontrivial and requires some knowledge about the structure of (extended) Pareto critical sets. In the following, we will briefly summarize the generic implications of the structural results mentioned in Sect. 2:

- According to [20], \mathcal{D}^∞ should (locally) be a differentiable manifold. In practice, this means that similar $\bar{x} \in \mathcal{D}_x^\infty$ should have similar $\bar{\alpha} \in \mathcal{D}_\alpha^\infty$.
- Following [17], for points $(\bar{x}, \bar{\alpha}) \in \mathcal{D}^\infty$ where \bar{x} lies on the edge of \mathcal{D}_x^∞ , we have to ensure that $\bar{\alpha}_j = 0$ for some $j \in \{1, \dots, k\}$. In particular, for multiple Pareto critical points on the same edge, the same component of the corresponding $\bar{\alpha}$ has to be zero.

According to our discussion in Sect. 3, we can generically not expect the smallest singular value of \mathcal{L} to be exactly zero in this case, as we will choose $N > \frac{k \cdot d}{n}$. Nonetheless, it turns out that if we use monomials as basis functions, the resulting space of polynomials is large enough to contain objective function vectors for nontrivial classes of infinite data sets.

Example 1 In this example, we will generate an MOP with two objective functions where the Pareto critical set is the unit circle \mathcal{S}^1 in \mathbb{R}^2 . To this end, let $N \in \mathbb{N}$ and

$$\bar{x}^j := \begin{pmatrix} \cos(2\pi \frac{j}{N}) \\ \sin(2\pi \frac{j}{N}) \end{pmatrix}, \quad j \in \{1, \dots, N\}.$$

The \bar{x}^j are points distributed equidistantly on \mathcal{S}^1 . While the choice of the \bar{x}^j is straightforward, the selection of the corresponding KKT vectors is less obvious. Since \mathcal{S}^1 has no edge (as defined in [17]), only the first one of the two structural results from above are relevant here, i.e., similar \bar{x} should have similar $\bar{\alpha}$ in the data. By construction, we have $\bar{x}^j \approx \bar{x}^{j+1}$ for $j \in \{1, \dots, N - 1\}$ and $\bar{x}^N \approx \bar{x}^1$, so the same should hold for $\bar{\alpha}^j$. One way of assuring this is to define the $\bar{\alpha}^j$ such that they periodically depend on j . For example, for the first component of $\bar{\alpha}^j$, we can choose $\cos(4\pi \frac{j}{N})$ as a periodic function in j and transform it such that it lies within $[0, 1]$. Since we must have $\bar{\alpha}^j \in \Delta_2$, this results in

$$\bar{\alpha}^j := \begin{pmatrix} 0.5(\cos(4\pi \frac{j}{N}) + 1) \\ 1 - 0.5(\cos(4\pi \frac{j}{N}) + 1) \end{pmatrix}, \quad j \in \{1, \dots, N\}. \tag{10}$$

Note that this choice is just one possibility and by no means unique. (In this case, we chose a different “frequency” for $\bar{\alpha}^j$ than for \bar{x}^j to avoid unwanted structures in the data.) The resulting data set for our algorithm is

$$\mathcal{D} := \{(\bar{x}^j, \bar{\alpha}^j) \in \mathbb{R}^2 \times \Delta_2 : j \in \{1, \dots, N\}\}.$$

We choose monomials up to degree 3 as basis functions, i.e.,

$$\mathcal{B} := \{x_1, x_1^2, x_1^3, x_2, x_1x_2, x_1^2x_2, x_2^2, x_1x_2^2, x_2^3\},$$

and use $N = 1000$ data points.

Figure 1a shows the sorted singular values of the matrix $\mathcal{L} \in \mathbb{R}^{2000 \times 18}$ resulting from the data and the basis functions. The first two singular values $s_1 = 3.92 \cdot 10^{-15}$ and $s_2 = 9.69 \cdot 10^{-15}$ are small, indicating that we found objective function vectors in the span of our basis functions for which the data set we constructed is exactly extended Pareto critical. There is an obvious gap from s_2 to $s_3 = 5.41$. Since s_1 and s_2 are both close to zero, we

choose the threshold $\bar{s} = s_2$, i.e., $I = \{1, 2\}$, in step 3 of Algorithm 1. The corresponding columns of V are given by

$$\begin{aligned} v_1 &= (-0.9040, 0, 0.3013, 0, 0, 0, 0, 0, 0.010, 0, 0, 0.3013, -0.030, 0, 0, 0, 0, 0.010)^\top, \\ v_2 &= (-0.030, 0, 0.010, 0, 0, 0, 0, 0, -0.3013, 0, 0, 0.010, 0.9040, 0, 0, 0, 0, -0.3013)^\top. \end{aligned}$$

In this example, it is easy to see that there is a certain pattern in v_1 and v_2 , so that we can write

$$\begin{aligned} &\text{span}(\{v_1, v_2\}) \\ &= \{(-3\sigma_1, 0, \sigma_1, 0, 0, 0, 0, 0, \sigma_2, 0, 0, \sigma_1, -3\sigma_2, 0, 0, 0, 0, \sigma_2)^\top : \sigma_1, \sigma_2 \in \mathbb{R}\}. \end{aligned} \tag{11}$$

Unfortunately, not all elements of $\text{span}(\{v_1, v_2\}) \setminus \{0\}$ in step 4 lead to desirable objective function vectors. To see this, consider the element c corresponding to $\sigma_1 = 0$ and $\sigma_2 = 1$, i.e.,

$$c = (0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, -3, 0, 0, 0, 0, 1)^\top.$$

The corresponding objective function vector is given by

$$\mathcal{F}(c)(x) = \begin{pmatrix} x_2^3 \\ x_2^3 - 3x_2 \end{pmatrix}. \tag{12}$$

For this objective function vector, the extended Pareto critical set indeed contains the given data set. However, the entire Pareto critical set for this problem is given by $\mathbb{R} \times [-1, 1]$, hence it contains significantly more than what we prescribed. In this case, the degeneracy is caused by the fact that this objective function vector does not depend on x_1 . A better choice for c would be, e.g., $\sigma_1 = 1$ and $\sigma_2 = 1$, resulting in

$$c = (-3, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, -3, 0, 0, 0, 0, 1)^\top.$$

The corresponding objective function vector $f = \mathcal{F}(c)$ is given by

$$f(x) = \begin{pmatrix} -3x_1 + x_1^3 + x_2^3 \\ -3x_2 + x_1^3 + x_2^3 \end{pmatrix}.$$

One can show that for this objective function vector, the KKT conditions are indeed equivalent to

$$\begin{aligned} x_1^2 + x_2^2 &= 1, \\ \alpha_1 &= x_1^2, \\ \alpha_2 &= x_2^2, \end{aligned}$$

i.e., the Pareto critical set is precisely S^1 with the corresponding KKT vectors given in (10). (In particular, we did not need to normalize c in step 4 of Algorithm 1 in this case.) Fig. 1b, c show the Pareto critical set and the image of the Pareto critical set under f .

Example 1 shows how the results from Sect. 3 can be used to derive an explicit expression for an objective function vector for a prescribed data set. The following example shows that we can even derive more general formulas.

Example 2 Using the same strategy as in Example 1, it is possible to numerically verify that arbitrary ellipses can be represented as Pareto critical sets of polynomial MOPs. For

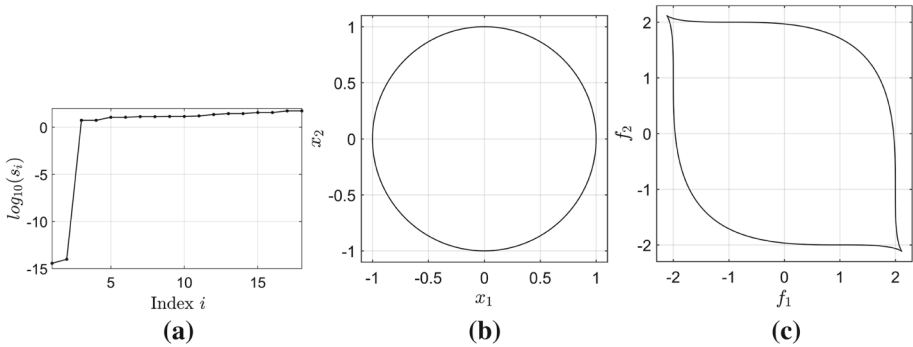


Fig. 1 **a** Singular values of \mathcal{L} in Example 1. **b** Pareto critical set of f . **c** Image of the Pareto critical set of f

$a, b \in \mathbb{R}^{>0}$, we merely have to replace the \bar{x}^j from Example 1 by

$$\bar{x}^j := \begin{pmatrix} a \cdot \cos(2\pi \frac{j}{N}) \\ b \cdot \sin(2\pi \frac{j}{N}) \end{pmatrix}.$$

In this case, if we consider the analogous expression to (11), we see that variations of a and b only influence a single component, respectively. In general, the following pattern can be recognized:

$$\text{span}(\{v_1, v_2\}) = \{(-3a^2\sigma_1, 0, \sigma_1, 0, 0, 0, 0, 0, \sigma_2, 0, 0, \sigma_1, -3b^2\sigma_2, 0, 0, 0, 0, \sigma_2)^\top : \sigma_1, \sigma_2 \in \mathbb{R}\},$$

which leads to the following conjecture: Let

$$f : \mathbb{R}^2 \rightarrow \mathbb{R}^2, \quad x \mapsto \begin{pmatrix} -3a^2x_1 + x_1^3 + x_2^3 \\ -3b^2x_2 + x_1^3 + x_2^3 \end{pmatrix}.$$

Then

$$P_c = \left\{ x \in \mathbb{R}^2 : \frac{x_1^2}{a^2} + \frac{x_2^2}{b^2} = 1 \right\}$$

and the KKT vector corresponding to $x \in P_c$ is given by $\alpha = \left(\frac{x_1^2}{a^2}, \frac{x_2^2}{b^2} \right)^\top$. After deriving this conjecture numerically, it is straight-forward to prove that it actually holds.

In Examples 1 and 2, the symbolic expressions could easily be verified. In particular, in step 4 of Algorithm 1 we were able to choose c such that the Pareto critical set did not contain more than what we intended, i.e., P_c was precisely the unit circle or an ellipse. This obviously only works if the data set is sufficiently well-structured. The following example shows a more complicated case.

Example 3 We are now searching for an MOP where the Pareto critical set contains three connected components $C_i, i = 1, 2, 3$, given by the following three non-intersecting straight lines:

$$C_i = p_i + [0, 1] \cdot \frac{1}{4} \frac{q_i}{\|q_i\|_2} \subseteq \mathbb{R}^2$$

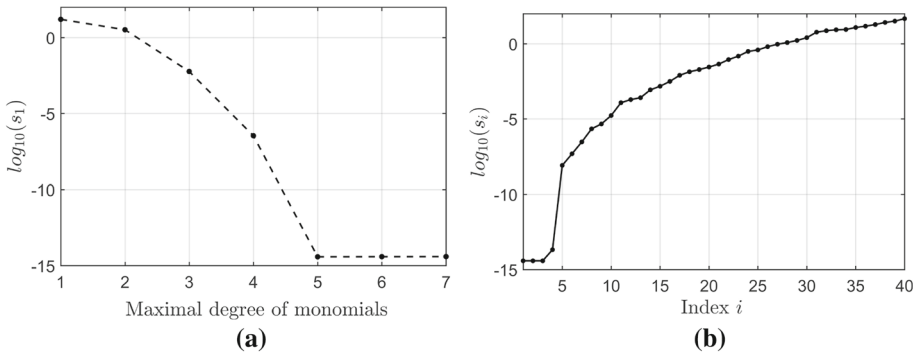


Fig. 2 **a** Smallest singular value of \mathcal{L} for different degrees of the monomial basis in Example 3. **b** Singular values of \mathcal{L} for degree 5

with

$$\begin{aligned}
 p_1 &= \begin{pmatrix} 0.15 \\ -0.20 \end{pmatrix}, \quad q_1 = \begin{pmatrix} 0.47 \\ 0.04 \end{pmatrix}, \quad p_2 = \begin{pmatrix} 0.47 \\ -0.32 \end{pmatrix}, \quad q_2 = \begin{pmatrix} 0.40 \\ 0.14 \end{pmatrix}, \\
 p_3 &= \begin{pmatrix} 0.37 \\ 0.18 \end{pmatrix}, \quad q_3 = \begin{pmatrix} 0.38 \\ 0.28 \end{pmatrix}.
 \end{aligned}$$

They are shown in Fig. 3a. For \mathcal{D}_x we choose $N_c = 500$ equidistant points on each C_i , the corresponding \mathcal{D}_α are chosen linearly from $(0, 1)^\top$ to $(1, 0)^\top$, and we again use monomials as basis functions. When dealing with more complex data sets, we first have to estimate the required degree of monomials for a satisfactory approximation. To this end, we repeat step 2 of Algorithm 1 for different maximal degrees. The smallest singular value depending on the maximal degree of the monomials is shown in Fig. 2a. We see that the monomials up to a degree of 5 are a promising choice, since the smallest singular values do not decrease further after that. Figure 2b shows all singular values for this set of basis functions. There is a relatively large gap from $s_4 = 2.09 \cdot 10^{-14}$ to $s_5 = 8.17 \cdot 10^{-9}$, suggesting that $\bar{s} = s_4$, i.e., $I = \{1, 2, 3, 4\}$, in step 3 of Algorithm 1 is a good choice. In this case, there is no obvious way to obtain an expression like (11) for $\text{span}(\{v_1, v_2, v_3, v_4\})$, which is why we choose $c = \frac{v_1}{\|v_1\|_2}$ in step 4. The Pareto critical set of $f = \mathcal{F}(c)$ and its image are shown in Fig. 3. As expected from the small singular values, the given data set is approximated almost perfectly. Unfortunately, we observe an additional connected component that is not contained in the data. Since we are unable to influence properties outside the given data set \mathcal{D} , additional Pareto critical points can be expected in the general case.

4.2 Inferring objectives from noisy data

In the previous examples, we assumed that we have precise data \mathcal{D} that we want to approximate with an extended Pareto critical set. However, there are many cases where this assumption is unrealistic, for instance real-world applications where the data stems from numerical simulations or measurements. Another example is stochastic multiobjective optimization, which we will consider here. We will only give a brief introduction on this topic and refer to [15] for a more detailed discussion.

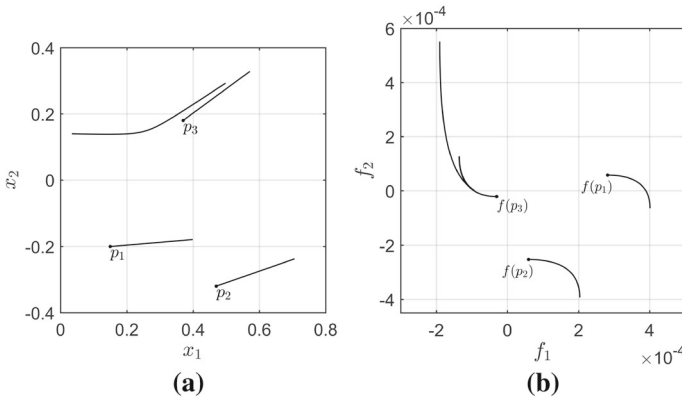


Fig. 3 **a** Pareto critical set of f in Example 3. **b** Image of the Pareto critical set of f

Let $\xi \in \mathbb{R}^m$ be a random vector and $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^k$. For $x \in \mathbb{R}^n$ let $\mathbb{E}[f(x, \xi)]$ be the (component-wise) expected value of $f(x, \xi)$. For $F(x) := \mathbb{E}[f(x, \xi)]$ we consider the stochastic multiobjective optimization problem

$$\min_{x \in \mathbb{R}^n} F(x). \tag{SMOP}$$

Since we cannot evaluate F directly, in practice the sample average

$$\tilde{f}^{N_s}(x) = \frac{1}{N_s} \sum_{j=1}^{N_s} f(x, \xi^j) \approx F(x) \tag{13}$$

is used, where ξ^1, \dots, ξ^{N_s} are independent and identically distributed samples of ξ . Using this approximation, we consider the *Sample Average Approximation* problem

$$\min_{x \in \mathbb{R}^n} \tilde{f}^{N_s}(x). \tag{SAA}$$

For $N_s = \infty$ the solutions of (SMOP) and (SAA) coincide. Otherwise, for a finite $N_s \in \mathbb{N}$, we can only expect the solution of (SAA) to be an approximation of the solution of (SMOP). In other words, we can consider the solution of (SAA) (together with the corresponding KKT vectors) as inexact data of the solution of the original problem (SMOP) and use our approach to approximate the original solution and objective function vector F . We illustrate this approach on the following Multiobjective Stochastic Location Problem from [15].

Example 4 Let $a := (-1, -1)^\top$ and $\xi := (\xi_1, 0)^\top$ be a random vector, where ξ_1 is uniformly distributed on $[0, 2]$. Let

$$f(x, \xi) := \begin{pmatrix} \|x - a\|_2^2 \\ \|x - \xi\|_2^2 \end{pmatrix}.$$

In this case, we have

$$F(x) = \mathbb{E}[f(x, \xi)] = \begin{pmatrix} \|x - a\|_2^2 \\ \|x - (1, 0)^\top\|_2^2 + 1/3 \end{pmatrix} = \begin{pmatrix} 2x_1 + x_1^2 + 2x_2 + x_2^2 + 2 \\ -2x_1 + x_1^2 + x_2^2 + 4/3 \end{pmatrix}, \tag{14}$$

so the Pareto critical (and in this case Pareto optimal) set of (SMOP) is given by the line connecting a and $(1, 0)^\top$. To obtain an approximation of (SMOP) in this case, we apply

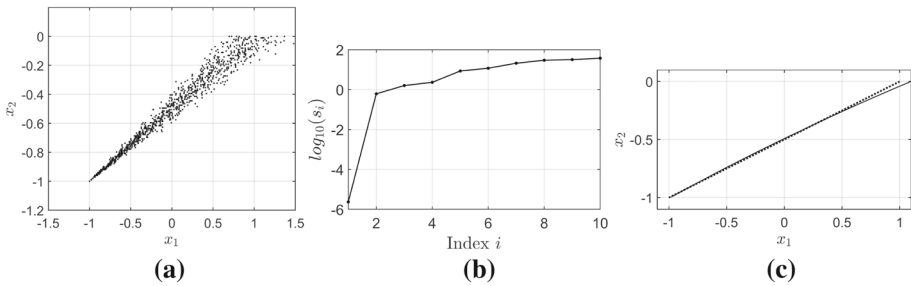


Fig. 4 **a** Approximation of the solution of (SMOP) with 1000 points for Example 4. **b** Singular values of \mathcal{L} . **c** Pareto critical set of the original problem (dotted line) and the approximation (solid line)

the weighting method to (SAA) for $N_s = 10$ with 100 equidistant weights in Δ^2 , and solve each of the resulting scalar problems 10 times (with different realizations of ξ_1). The result is shown in Fig. 4a. Since there is no noise in the first component of f , the approximation is relatively accurate close to a and becomes worse when moving towards $(1, 0)^\top$.

We now interpret the points in Fig. 4a as the data set \mathcal{D}_x . The KKT vector $\alpha \in \mathcal{D}_\alpha$ corresponding to $x \in \mathcal{D}_x$ is chosen as the weight in the weighting method that was used to compute x (cf. Sect. 5). We choose \mathcal{B} as the monomials up to degree 2, i.e.,

$$\mathcal{B} := \{x_1, x_1^2, x_2, x_1x_2, x_2^2\}.$$

In this basis, the objective function vector F of (SMOP) can be represented exactly (up to the constants in both components) by the coefficient vector

$$\bar{c} = (2, 1, 2, 0, 1, -2, 1, 0, 0, 1)^\top. \tag{15}$$

When applying Algorithm 1, we obtain the singular values shown in Fig. 4b. The objective function vector $x \mapsto (2x_2 + x_2^2, x_2^2)^\top$ corresponding to the smallest singular value $s_1 = 2.34 \cdot 10^{-6}$ is degenerate due to the missing dependency on x_1 . The next smallest singular values are

$$\begin{aligned} s_2 &= 0.6134, \\ s_3 &= 1.5927, \\ s_4 &= 2.3106, \\ s_5 &= 8.5384. \end{aligned}$$

Hence, due to the gap from s_4 to s_5 , we choose $\bar{s} = s_4$, i.e., $I = \{1, 2, 3, 4\}$, in step 3. Calculating a (normalized) sparse basis $\{w_1, w_2, w_3, w_4\}$ of $\text{span}(\{v_1, v_2, v_3, v_4\})$ results in

$$\begin{aligned} w_1 &= (0, 0, -1, 0, -0.5, 0, 0, 0, 0, -0.5)^\top, \\ w_2 &= (-0.4979, 0.2505, -0.9981, -1, 0, 0, -0.0021, 0, 0.0078, -0.9965)^\top, \\ w_3 &= (-1, -0.5077, 0, -0.0040, 0, 0.9607, -0.4397, -0.0018, 0, 0.0008)^\top, \\ w_4 &= (0.7279, -0.1351, 0, 1, -0.4932, -0.0001, 0, -0.4782, 0.4424, 0.0001)^\top, \end{aligned}$$

which shows that in step 4, we can choose

$$c^* := -2w_1 - 2w_3 = (2, 1.0155, 2, 0.0080, 1, -1.9214, 0.8794, 0.0037, 0, 0.9985)^\top,$$

which is close to \bar{c} (cf. (15)). The Pareto critical set of the corresponding objective function vector $\mathcal{F}(c^*)$ is shown in Fig. 4c. A numerical approximation of the Hausdorff distance

between the two sets (using a pointwise discretization) yields $9 \cdot 10^{-2}$ (and the corresponding points of maximal distance are located close to $(1, 0)^\top$). As functions, comparing F and $\mathcal{F}(c^*)$ (up to constants, cf. (14)) around the Pareto critical set yields

$$\max_{x \in [-1.1, 1.1] \times [-1.1, 0.1]} \|(F(x) - (2, 4/3)^\top) - \mathcal{F}(c^*)(x)\|_\infty \approx 2.38 \cdot 10^{-1},$$

showing that we were able to construct a good approximation of the objective function vector F from noisy data.

In all examples we considered so far, both the variable space and the image space were two-dimensional, i.e., we searched for an objective function vector $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$. In the following example, we consider a higher-dimensional case which is inspired by Example 1 in [34].

Example 5 Let $\xi \in \mathbb{R}$ be a random variable, uniformly distributed on $[0, 2]$. For

$$\begin{aligned} a^1 &:= (1, \dots, 1) \in \mathbb{R}^{10} \\ a^2 &:= (-1, \dots, -1) \in \mathbb{R}^{10} \\ a^3 &:= (\xi, -1, 1, -1, \dots, 1, -1) \in \mathbb{R}^{10} \end{aligned}$$

consider the function $f : \mathbb{R}^{10} \times \mathbb{R} \rightarrow \mathbb{R}^3$ with

$$f_i(x, \xi) := (x_i - a_i^i)^4 + \sum_{j=1, j \neq i}^n (x_j - a_j^i)^2, \quad i \in \{1, 2, 3\}. \tag{16}$$

Note that only f_3 depends on ξ and we have

$$\mathbb{E}[f_3(x, \xi)] = f_3(x, 1) + \frac{1}{3}.$$

To generate the data for our inverse algorithm, we apply the weighting method to (SAA) as in Example 4. In this case, we use $N_s = 100$ samples with 351 evenly distributed weights in Δ^3 and solve each of the resulting scalar problems 5 times. The resulting 1755 points, projected onto the first three components, are shown in Fig. 5a. As data for the KKT vectors, we again use the corresponding weighting parameters in the weighting method. As basis functions, we use the union of the monomials in 10 variables up to degree 2 and $\{x_1^3, x_1^4, x_2^3, x_2^4, x_3^3, x_3^4\}$, such that the function $F(x) = \mathbb{E}[f(x, \xi)]$ is again contained in the span of the basis functions (up to constants). Let $\bar{c} \in \mathbb{R}^{k \cdot d} = \mathbb{R}^{3 \cdot 71} = \mathbb{R}^{213}$ be the vector of coefficients such that $F = \mathcal{F}(\bar{c})$. The singular values of $\mathcal{L} \in \mathbb{R}^{17550 \times 213}$ when applying Algorithm 1 are shown in Fig. 5b. The first gap in the singular values occurs from $s_{65} = 6.0154 \cdot 10^{-5}$ to $s_{66} = 0.0785$. The objective function vectors corresponding to s_i for $i \in \{1, \dots, 65\}$ are all degenerate, since they are almost constant with respect to x_1 , similar to Example 4.

The second visible gap is between $s_{68} = 0.0850$ and $s_{69} = 0.2928$. Thus, we will first try $\bar{s} = 10^{-0.9} = 0.1259$ as a threshold in Algorithm 1, which is indicated by the lower dashed line in Fig. 5b. Let $\{v_1, \dots, v_{68}\} \subseteq \mathbb{R}^{213}$ be the set of corresponding right-singular vectors and let $V \in \mathbb{R}^{213 \times 68}$ be the matrix with columns v_1, \dots, v_{68} . In Example 4, we were able to reconstruct the exact coefficient vector \bar{c} from the span of $\{v_1, \dots, v_{68}\}$ “by hand”. Due to the higher complexity of this example, we need a more sophisticated strategy here, which involves exploitation of the structure of (16). To this end, note that ξ only appears in the coefficients for f_3 of the 13 basis functions containing x_1 , and all other coefficients are fixed. Let $J \subseteq \{1, \dots, 213\}$ be the set of indices of these fixed coefficients, let $V_J \in \mathbb{R}^{200 \times 68}$ be the

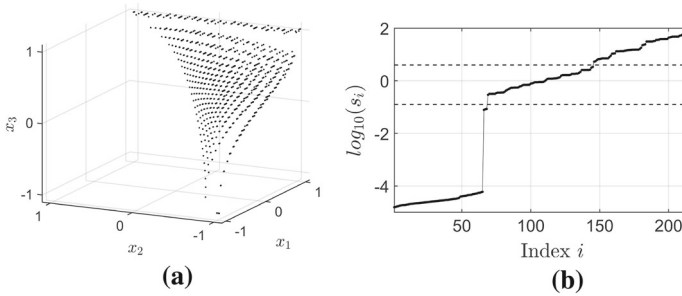


Fig. 5 **a** Approximation of the solution of (SMOP) with 1755 points for Example 5. **b** Singular values of \mathcal{L} . The dashed lines indicate the different thresholds that were used in Example 5

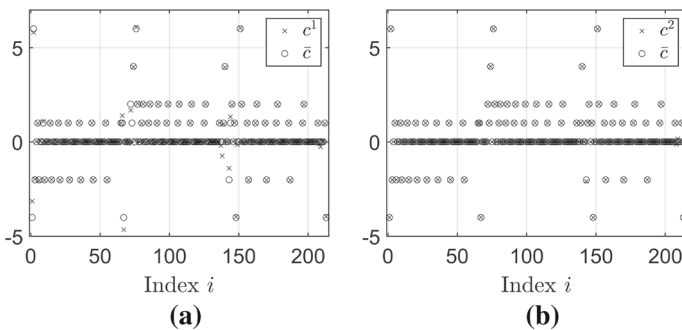


Fig. 6 Approximation of the original coefficient vector $\bar{c} \in \mathbb{R}^{213}$ (with $\mathcal{F}(\bar{c}) = F$) using a threshold of **a** $\bar{s} = -0.9$ and **b** $\bar{s} = 0.6$

submatrix of V containing only rows with an index in J and let $\bar{c}_J \in \mathbb{R}^{200}$ be the vector of fixed coefficients. Let $\sigma \in \mathbb{R}^{68}$ be a least squares solution of the linear system

$$V_J \sigma = \bar{c}_J.$$

Then $c^1 = V\sigma$ yields as an approximation of \bar{c} , shown in Fig. 6a. When comparing c^1 to \bar{c} , we have

$$\|c^1 - \bar{c}\|_\infty = 1.0794, \quad \frac{1}{213} \sum_{i=1}^{213} |c_i^1 - \bar{c}_i| = 0.0426.$$

So while the average error of c^1 is relatively small, there are some outliers where the error is large.

To improve the quality of the approximation, we will now apply the same strategy again, but this time with a threshold of $\bar{s} = 10^{0.6} = 3.9811$, corresponding to the third gap of the singular values in Fig. 5b from $s_{145} = 3.3666$ to $s_{146} = 5.3254$. The resulting approximation c^2 is shown in Fig. 6b, where it is almost indistinguishable from \bar{c} . The maximal and average errors are

$$\|c^2 - \bar{c}\|_\infty = 0.1536, \quad \frac{1}{213} \sum_{i=1}^{213} |c_i^2 - \bar{c}_i| = 0.0023,$$

confirming the observation.

Unfortunately, in the previous example, we were only able to infer the (coefficients of the) original objective function vector by exploiting some of the structure of this specific function. In general, when considering the span of the right-singular vectors corresponding to some gap in the singular values, it is difficult (or even impossible) to reconstruct the original coefficient vector without taking further steps such as a structural assumption. This will be further discussed in the conclusion.

5 Application 2: Generation of surrogate models of expensive MOPs

In this section, we will use the results from Sect. 3 for the generation of surrogate models for MOPs with an objective function vector f^e that is known but very costly to evaluate. This scenario occurs frequently for complex physics simulations, e.g., when the system under consideration is described by a partial differential equation, cf. [2,3,25,29] for examples. Here, while it is often possible to calculate single Pareto critical points, the computation of the full Pareto critical set via a fine pointwise approximation is computationally infeasible. In this situation, we will use a few Pareto critical points of the expensive model f^e and their corresponding KKT vectors as data points. Our goal is to find an MOP whose extended Pareto critical set is as close as possible to the extended Pareto critical set of f^e while using as few data points as possible.

When computing solutions of the expensive model for the generation of the data for our approach, it is important to not only obtain the Pareto critical points in the variable space, but also the corresponding KKT vectors. Fortunately, there are common methods where this is easy to achieve:

- Weighting method: If $x^* = \arg \min_{x \in \mathbb{R}^n} \sum_{i=1}^k \alpha_i^* f_i(x)$ for some weighting vector $\alpha^* \in \Delta_k$, then α^* is a KKT vector of x^* (by first-order optimality).
- ε -constraint method (cf. [27]): For $l \in \{1, \dots, k\}$ and $\varepsilon_j \in \mathbb{R}$, $j \in \{1, \dots, k\} \setminus \{l\}$, let x^* be the solution of

$$\begin{aligned} & \min_{x \in \mathbb{R}^n} f_l(x) \\ & \text{s.t. } f_j(x) \leq \varepsilon_j, \quad j \in \{1, \dots, k\} \setminus \{l\}. \end{aligned}$$

By the first-order optimality conditions for this problem, there are $\mu_j^* \geq 0$, $j \in \{1, \dots, k\} \setminus \{l\}$, such that

$$\nabla f_l(x^*) + \sum_{j \neq l} \mu_j^* \nabla f_j(x^*) = 0.$$

Let

$$\alpha^* = \frac{1}{\alpha_1 + \dots + \alpha_k} \alpha \quad \text{for } \alpha_i = \begin{cases} \mu_i^*, & i \neq l, \\ 1, & i = l. \end{cases}$$

Then α^* is a KKT vector of x^* .

- Reference point method (cf. [27]): Let $z \in \mathbb{R}^k$ and x^* be a solution of

$$\min_{x \in \mathbb{R}^n} \|f(x) - z\|^2$$

with $z_i \leq f_i(x^*)$ for all $i \in \{1, \dots, k\}$ and $z \neq f_i(x^*)$. By the first-order optimality condition for this problem, we have

$$0 = \nabla(x \mapsto \|f(x) - z\|^2)(x^*) = 2Df(x^*)^\top (f(x^*) - z).$$

By assumption, $f_i(x^*) - z_i \geq 0$ for all $i \in \{1, \dots, k\}$ and $\sum_{i=1}^k f_i(x^*) - z_i > 0$, so we obtain a KKT vector via

$$\alpha^* = \frac{f(x^*) - z}{\sum_{i=1}^k f_i(x^*) - z_i}.$$

For methods which are not directly related to the KKT conditions of the MOP, like evolutionary algorithms, KKT vectors are more difficult to obtain. Given only the Pareto critical (or optimal) point, a straight-forward way to obtain the corresponding KKT vector would be to evaluate the gradients of the objective functions and solve (KKT) as a linear system in α . However, this approach can obviously be very time consuming. Furthermore, knowledge of the derivatives of the expensive model is required. A much cheaper alternative is to exploit the fact that KKT vectors are orthogonal to the linearized Pareto front [20]. For a pointwise approximation of the Pareto front, e.g., obtained by NSGA-II, we can use linear regression in each point of the front using only the neighboring points on the front to obtain an approximation of the tangent space of the Pareto front. While this requires a relatively even discretization of the Pareto front, it is much cheaper than assembling and solving the above-mentioned linear system.

Surrogate modeling is a very active area of research and has been used extensively for simulation and optimization, see [4,33] for overviews. In recent years, surrogate models have also attracted interest in the multiobjective optimization community. All methods proposed so far have the common goal of finding a surrogate model for the objective function f^e , for instance by polynomial regression (cf., e.g., [8,36]). Consequently, the surrogate model will possess a dominance relation similar to the original function and as a result, dominance-based methods like evolutionary algorithms can be applied. In contrast to this, our approach constructs surrogate models which resemble the original KKT conditions. This means that we (in general) do not obtain a surrogate model for the objective function but for the first-order optimality condition such that KKT-based methods like continuation [34] can be used.

When “fitting” a surrogate model to a data set of limited size as in this case, it is important avoid *underfitting* and *overfitting*. These terms are common in statistics and machine learning, but they apply here in a similar fashion. In general, underfitting means that the chosen model is not able to capture all structures that are present in the data set. In our context, this means that we chose an unsuited (e.g., too small) set of basis functions. When using monomials as basis functions, one can try to circumvent this by using a higher maximal degree (as in Example 3). On the other hand, overfitting means that the model captures structures in the data set that were caused by noise and are highly dependent on the data used for fitting the model. In our context, this happens when the number of basis functions d in \mathcal{B} is too large. A necessary condition to circumvent this is to ensure that $n \cdot N \geq k \cdot d$, i.e.,

$$d \leq \frac{n \cdot N}{k}. \tag{17}$$

As discussed in Sect. 3, if this condition does not hold then we always find an objective function vector in the chosen basis where the data points are exactly extended Pareto critical. Thus, if (17) is violated, overfitting is unavoidable.

To illustrate the behavior of our method, we begin with an example where the objective function vector is cheap to evaluate and we already know the solution of the MOP. We consider the problem $L \& H_{2 \times 2}$ from [18], where the objective function vector is non-polynomial and has a complex (extended) Pareto critical set.

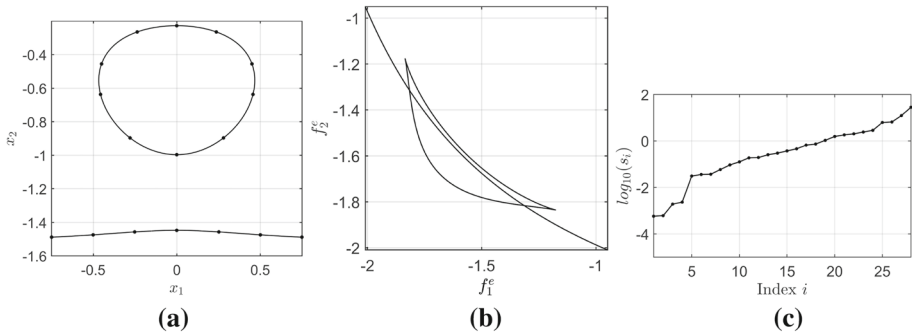


Fig. 7 **a** Pareto critical set of $(L\&H_{2 \times 2})$. The dots represent the $N = 17$ data points used for the surrogate model construction. **b** Image of the Pareto critical set. **c** Singular values of \mathcal{L} for monomials up to degree 4 for the chosen data points

Example 6 Consider the MOP

$$\begin{aligned} & \min_{x \in \mathbb{R}^2} f^e(x) && (L\&H_{2 \times 2}) \\ & s.t. \quad x \in [-0.75, 0.75] \times [-2.5, 0.12] \end{aligned}$$

for

$$f^e(x) := - \begin{pmatrix} \frac{\sqrt{2}}{2}x_1 + \frac{\sqrt{2}}{2}b(x) \\ -\frac{\sqrt{2}}{2}x_1 + \frac{\sqrt{2}}{2}b(x) \end{pmatrix}$$

with

$$\begin{aligned} b(x) &:= 0.2g(x, (0, 0)^\top, 0.65) + 1.5g(x, (0, -1.5)^\top, 2.8), \\ g(x, p_0, \sigma) &:= \sqrt{\frac{2\pi}{\sigma}} \exp\left(-\frac{\|x - p_0\|_2^2}{\sigma^2}\right). \end{aligned}$$

The Pareto critical set of this MOP and its image are shown in Fig. 7a, b, respectively. (Note that since all Pareto critical points on the boundary of the feasible set are also Pareto critical for the unconstrained problem, we can consider this problem as unconstrained.) For the surrogate model construction, we choose the $N = 17$ data points depicted in Fig. 7a. We choose all monomials up to degree 4 as basis functions. The reason for this choice is that for larger degrees we have $|\mathcal{B}| = d \geq 20$ such that $n \cdot N = 34 < 40 \leq k \cdot d$, which would result in overfitting.

The surrogate model is now constructed from the data set by applying Algorithm 1. The singular values of \mathcal{L} are shown in Fig. 7c. In steps 3 and 4, we choose the coefficient vector $c = \frac{v_1}{\|v_1\|_2}$ corresponding to the smallest singular value $s_1 = 5.76 \cdot 10^{-4}$. A comparison between the Pareto critical sets of the corresponding objective function vector $f = \mathcal{F}(c)$ and the original objective function vector $(L\&H_{2 \times 2})$ is shown in Fig. 8. We see in (a) that the Pareto critical sets are almost identical besides the two additional connected components at the top. After filtering these out (e.g., by applying clustering algorithms, cf. [32]), the Hausdorff distance between the two sets is $4 \cdot 10^{-3}$. Figure 8b shows the image of the Pareto critical set of f under the original objective function vector f^e without the additional connected components. Similar to the variable space, the Pareto fronts are almost identical with a Hausdorff distance of $1.6 \cdot 10^{-3}$.

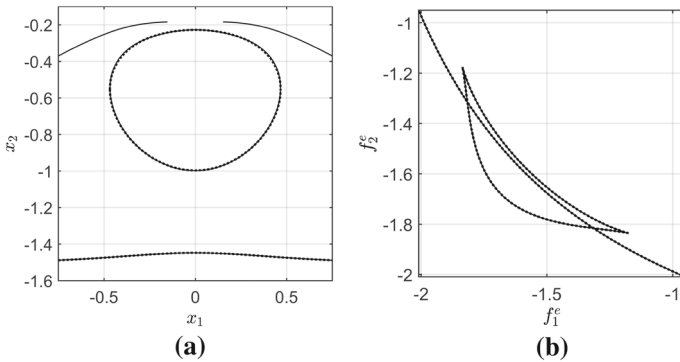


Fig. 8 **a** The Pareto critical sets of $(L \& H_{2 \times 2})$ (dotted line) and its approximation f (solid line). **b** The images of the Pareto critical sets of $(L \& H_{2 \times 2})$ (dotted line) and f under f^e (solid line)

The previous example shows that few data points of the original objective function vector can suffice to generate a good surrogate model, even if the original objective function vector does not lie in the span of the chosen basis functions. In order to highlight the potential for increased efficiency in real-world applications, our next example considers an MOP where the evaluation of the objective function vector is very expensive.

Example 7 In this example, we consider the flow around a cylinder governed by the 2D incompressible Navier–Stokes equations at a Reynolds number of 100, where the goal is to influence the flow field by rotating the cylinder (cf. Fig. 9a):

$$\begin{aligned}
 \dot{y}(x, t) + y(x, t) \cdot \nabla y(x, t) &= \nabla p(x, t) + \frac{1}{Re} \Delta y(x, t), \\
 \nabla \cdot y(x, t) &= 0, \\
 y(x, 0) &= y^0(x).
 \end{aligned}
 \tag{NSE}$$

Here, y is the flow velocity and p is the pressure. For the non-rotating cylinder, the well-known *von Kármán vortex street* occurs. This is a periodic solution where vortices detach alternately from the upper and lower edge of the cylinder, respectively. This setup is a classical problem from flow control which has been studied extensively in the literature both using direct approaches as well as surrogate models, see [29] and the references therein. The classical goal is to stabilize the flow, i.e., to minimize the vertical velocity. This can be associated with minimizing the vertical force on the cylinder, the *lift* C_L . As a second goal, we want to minimize the control effort, which results in the following multiobjective optimal control problem:

$$\begin{aligned}
 \min_{u \in L^2([t_0, t_e], \mathbb{R})} & \left(\int_{t_0}^{t_e} C_L^2(t) dt \right) \\
 & \left(\int_{t_0}^{t_e} u^2(t) dt \right) \\
 \text{s.t.} & \quad \text{(NSE)}.
 \end{aligned}
 \tag{18}$$

By introducing a sinusoidal control $u(t) = x_1 \sin(2\pi x_2 t)$ and assuming that the control-to-state mapping is injective, Problem (18) can be transformed into the MOP

$$\min_{x \in \mathbb{R}^2} f^e(x) \quad \text{with} \quad f^e(x) := \left(\int_{t_0}^{t_e} C_L^2(t) dt, \int_{t_0}^{t_e} (x_1 \sin(2\pi x_2 t))^2 dt \right).
 \tag{19}$$

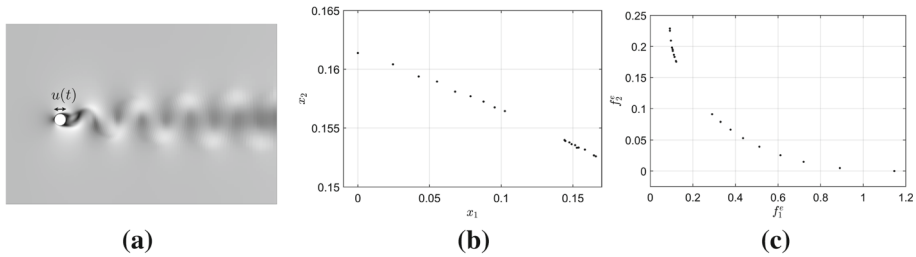


Fig. 9 **a** Flow around a cylinder, controlled via cylinder rotation. **b** Result of the weighting method applied to the MOP (19) in the variable space. **c** Image of the resulting points under the objective function (19)

Since the Navier–Stokes equations are a system of nonlinear partial differential equations, we have to introduce a spatial discretization (here via the finite volume method) with 22, 000 cells, which results in 66, 000 degrees of freedom at each time instant. Consequently, it is infeasible to accurately solve Problem (19) directly, regardless of the used method.

One way to approach this problem is to introduce a surrogate model for the system dynamics (NSE), for instance via *Proper Orthogonal Decomposition* [29]. In contrast to this, here, we directly construct a surrogate model for the MOP (19) instead of the system dynamics. In order to generate the required data points \mathcal{D} , we apply scalarization via the weighting method (i.e., $\min w_1 f_1^e + w_2 f_2^e$) to (19) with varying weights

$$w^i = \left(\frac{i - 1}{25}, 1 - \frac{i - 1}{25} \right)^T, \quad i \in \{1, \dots, 26\}. \tag{20}$$

An advantage of the weighting method is that we directly obtain the KKT vectors of the resulting Pareto optimal points as the corresponding weights that were used to calculate them. Since there are convergence issues for $i \in \{10, \dots, 16\}$ (likely due to a large number of local minima, which is a known problem), we will exclude these points from our data set. The remaining 19 points are shown in Fig. 9b, c. Considering \mathcal{D}_x and \mathcal{D}_α , it appears that the Pareto set consists of a single one-dimensional connected component whose corresponding KKT vectors are monotonically increasing and decreasing in their first and second component, respectively. Due to this simple structure, we take all monomials up to degree 2 as our set of basis functions. The singular values of the resulting $\mathcal{L} \in \mathbb{R}^{38 \times 10}$ are shown in Fig. 10a. The smallest singular values $s_1 = 2.82 \cdot 10^{-4}$ and $s_2 = 5.95 \cdot 10^{-4}$ correspond to objective function vectors where the influence of x_1 is relatively small. (In particular, the hessian matrices of both objective functions in both objective function vectors are almost singular.) Therefore, the corresponding Pareto critical sets are degenerate similar to the objective function vector (12) in Example 1. Due to this, we instead consider the objective function vector corresponding to the third singular value $s_3 = 5.96 \cdot 10^{-3}$ as our surrogate model, given by

$$f(x) = \begin{pmatrix} -0.0519x_1^2 - 0.9285x_1x_2 + 0.1588x_1 + 0.1542x_2^2 + 0.1046x_2 \\ -0.0136x_1^2 - 0.2704x_1x_2 + 0.0437x_1 + 0.0054x_2^2 - 0.0008x_2 \end{pmatrix}. \tag{21}$$

A projection of the corresponding extended Pareto critical set is depicted in Fig. 10b, showing that all data points are close to the solution of the surrogate problem. In order to obtain an approximation of the Pareto front of the original MOP (19), we can evaluate the original objective function vector f^e in a pointwise discretization of the Pareto critical set of the surrogate model f . In order to evaluate the performance, we compare our results with the well-known NSGA-II algorithm [11] (implementation from MATLAB’s Global Optimization

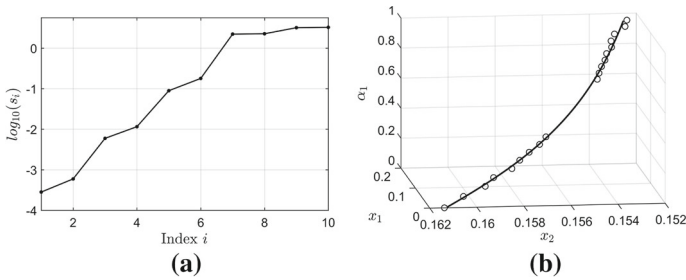


Fig. 10 **a** Singular values of \mathcal{L} in Example 7. **b** Pareto critical set of (21) (solid line) and the data in \mathcal{D}_x (circles), where the first KKT multiplier α_1 is shown in the third dimension

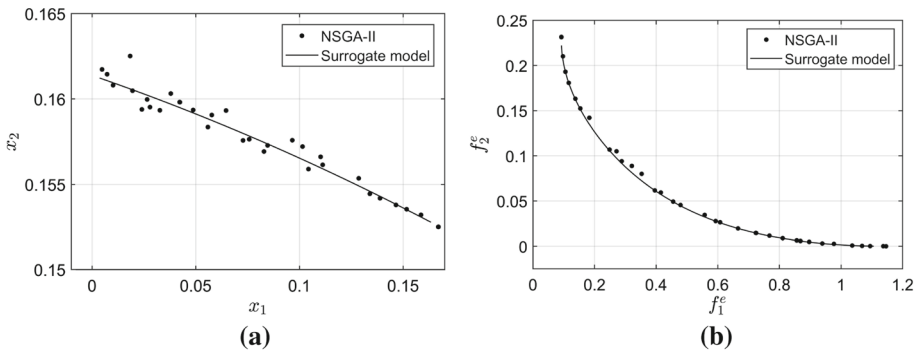


Fig. 11 **a** The approximation of the Pareto set via the surrogate model compared to NSGA-II in Example 7. **b** Comparison of the corresponding approximations of the Pareto fronts

Toolbox) directly applied to the MOP (19). The results are depicted in Fig. 11. Here, we have used an initial population size of 100 for NSGA-II and a discretization of the Pareto critical set of our surrogate model (21) with 468 equidistant points. Figure 11 shows that although we only used 19 data points for the generation of our surrogate model and there was a gap in our data set, we are able to obtain a good approximation of the Pareto set and front in a very efficient manner.

Example 8 In order to discuss possible challenges with higher-dimensional problems, we consider a multiobjective optimal control problem for autonomous electric vehicles, see [13,30] for a detailed description. The problem is to set the acceleration of a nonlinear model for the longitudinal velocity of an electric vehicle – consisting of ordinary differential equations for both the mechanical system and the electronic components – in such a way that the vehicle drives both fast and comfortably on a given track:

$$\begin{aligned}
 \min_{u \in L^2([t_0, t_e], \mathbb{R})} & \begin{pmatrix} -\int_{t_0}^{t_e} y_1(t) dt \\ \|\dot{y}\|_{L^2} \end{pmatrix} \\
 \text{s.t. } & \dot{y} = g(y, u), \\
 & u(t) \in [-500, 1000].
 \end{aligned} \tag{22}$$

Here, $y = [v, S, u_{dL}, u_{dS}]$ is the system state comprised of the vehicle velocity, the battery state of charge, and the long and short term voltage drops, respectively [13]. Consequently,

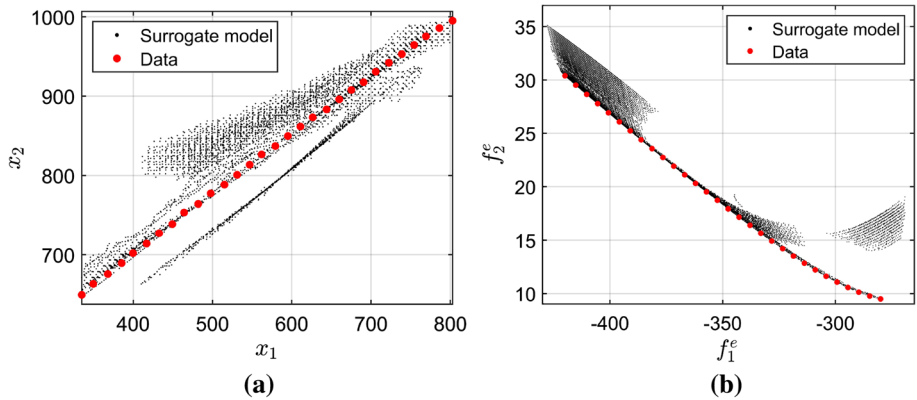


Fig. 12 **a** Projection of the Pareto critical set of the surrogate model and the data that was used to create it for Example 8. **b** Image of the Pareto critical set of the surrogate model and the data under the original objective function vector

the two objectives are maximizing the final position and minimizing the integrated acceleration, representing fast and comfortable driving. A discretization of the control input u into four piecewise constant parts over the time horizon $t_e - t_0 = 20$ (represented via $x \in [-500, 1000]^4 \subseteq \mathbb{R}^4$) results in the objective function vector $f^e : \mathbb{R}^4 \rightarrow \mathbb{R}^2$.

We use the ε -constraint method to generate 30 data points for our inverse approach. (Note that although this is a constrained MOP, the data points were computed such they do not lie on the boundary of the feasible set.) As derived in the beginning of this section, we obtain the KKT vectors via the optimality conditions of the scalarized problems. As basis functions, we use monomials up to degree 3. Projections of the Pareto critical set of the resulting surrogate model (approximated via the Continuation Method) and the data on which it is based are shown in Fig. 12a. Although the Pareto critical set contains the data, it contains additional points which are unrelated to the data. Figure 12b shows the image of the Pareto critical set under the original objective function vector f^e and the image of the data. In accordance with the variable space, the image of the Pareto critical set of the surrogate model covers the true Pareto front, but it also contains points which are clearly not Pareto optimal.

6 Conclusion and outlook

In this article, we present a way to construct an objective function vector of an MOP such that its extended Pareto critical set contains a given data set. This is realized by considering the x^* and α^* in the KKT conditions as given by the data and then searching for an objective function vector $f \in C^1(\mathbb{R}^n, \mathbb{R}^k)$ that solves the resulting system of equations. By using a finite set of basis functions $\mathcal{B} \subseteq C^1(\mathbb{R}^n, \mathbb{R})$, f can be obtained via singular value decomposition, which results in Algorithm 1.

The ability to infer objective function vectors from (potentially noisy) data has several potential applications. In examples, we showed how it can be used to generate test problems for solution methods of MOPs and to approximate the Pareto set and objective function vector of stochastic MOPs. Furthermore, the approach can be used to significantly reduce the computational effort for expensive MOPs. Using several data points from the expensive problem, a much cheaper surrogate model can be constructed which can be solved significantly faster.

While we demonstrate the validity of the proposed approach in several examples, we emphasize that there are open problems that should be addressed in order to increase the range of applicability, in particular for real-life scenarios. These will be discussed in the following.

- As explained in the introduction, the assumption that KKT vectors are available in the data is strong. While we found multiple examples where this data is available, it still restricts the range of applications. Thus, to make our method more applicable, this assumption should be weakened. In [12], this assumption was avoided by approximating the Pareto set of a parameter-dependent MOP for different parameters using a finite number of solutions obtained via the weighting method. This idea can not directly be conveyed to our non-convex case, as the weighting method generally does not yield all Pareto optimal (or critical) points. Instead, knowledge about structural properties of Pareto critical sets and the corresponding KKT vectors might be of use [17].
- For the generation of surrogate models, it is important to ensure that the (extended) Pareto critical set of the surrogate model is indeed a good approximation of the actual (extended) Pareto critical set. The convergence result in Theorem 2 states that the smallest singular value of \mathcal{L} is an upper bound for the Euclidean norm of the KKT conditions in the data points. However, this can not directly be used to obtain an estimate for the Hausdorff distance between the actual Pareto critical set and its surrogate approximation. Furthermore, even if the data is exactly contained in the Pareto critical set of the surrogate model, the surrogate model will generally have additional Pareto critical points which are not contained in the data. This can be seen in Examples 3, 6 and 8. Thus, a strategy needs to be developed to exclude undesired additional points.
- In step 4 of Algorithm 1, one has to choose an element in the span of certain right-singular vectors of \mathcal{L} . In general, depending on the chosen threshold, this span can be large. For instance, the span corresponding to the largest chosen threshold in Example 5 is 145-dimensional. Furthermore, different elements in this span can produce objective function vectors with significantly differing Pareto critical sets, even if they contain the data. If no additional information outside of the data set is given, then there is no obvious way to choose an element in the span. Thus, as already mentioned in Remark 1, it makes sense to think about additional criteria in step 4 that render the resulting function more well-behaved or “regular” in an appropriate sense.
- By construction, the data points in \mathcal{D}_x are not necessarily Pareto optimal for the objective function vectors that result from Algorithm 1. Thus, methods which can specifically compute Pareto critical sets have to be used when working with the inferred objective functions vectors. In this article, we used the Continuation Method from [34]. While this works well for the low-dimensional examples we presented here, it is increasingly challenging in high-dimensional cases. Thus, our inverse approach could be improved by assuring that the data points are actually Pareto optimal and not just Pareto critical. As sufficient optimality conditions for MOPs use second order derivatives (cf. [27]), a possible way to control the optimality of the data set might be to incorporate the Hessians of the basis functions in our approach.
- Since the variables in most MOPs in practical applications have to satisfy certain constraints, our method should be extended to constrained MOPs.
- For the reasons mentioned at the end of Sect. 3, we mainly used monomials up to different maximal degrees as basis functions \mathcal{B} . Although this lead to satisfactory results in the examples considered here, there might be more sophisticated choices, in particular if one has some knowledge of the problem structure. Moreover, promising approaches for

dictionary learning have recently been proposed, for instance in the context of dynamical systems approximations [28].

Acknowledgements This research was funded by the DFG Priority Programme 1962 “Non-smooth and Complementarity-based Distributed Parameter Systems”.

Funding Open Access funding enabled and organized by Projekt DEAL.

Compliance with ethical standards

Data Availability: The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

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/>.

References

1. Ahuja, R.K., Orlin, J.B.: Inverse optimization. *Oper. Res.* **49**(5), 771–783 (2001)
2. Albunni, M.N., Rischmuller, V., Fritzsche, T., Lohmann, B.: Multiobjective optimization of the design of nonlinear electromagnetic systems using parametric reduced order models. *IEEE Trans. Magn.* **45**(3), 1474–1477 (2009)
3. Beermann, D., Dellnitz, M., Peitz, S., Volkwein, S.: Set-oriented multiobjective optimal control of PDEs using proper orthogonal decomposition. In: Winfried, K., Anja, M., Stefan, V. (eds.) *Reduced-Order Modeling (ROM) for Simulation and Optimization*, pp. 47–72. Springer (2018)
4. Benner, P., Gugercin, S., Willcox, K.: A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Rev.* **57**(4), 483–531 (2015)
5. Brunton, S. L., Proctor, J. L., Kutz, J. N.: Discovering governing equations from data by sparse identification of nonlinear dynamical systems. In: *Proceedings of the National Academy of Sciences* p. 201517384 (2016)
6. Chan, T.C.Y., Craig, T., Lee, T., Sharpe, M.B.: Generalized inverse multiobjective optimization with application to cancer therapy. *Oper. Res.* **62**(3), 680–695 (2014)
7. Chan, T.C.Y., Lee, T.: Trade-off preservation in inverse multi-objective convex optimization. *Eur. J. Oper. Res.* **270**(1), 25–39 (2018)
8. Chugh, T., Sindhya, K., Hakanen, J., Miettinen, K.: A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms. *Soft Comput.* **23**, 3137–3166 (2017)
9. De Melo, W.: On the structure of the Pareto set of generic mappings. *Bull. Braz. Math. Soc.* **7**(2), 121–126 (1976)
10. Deb, K.: Multi-objective genetic algorithms: problem difficulties and construction of test problems. *Evol. Comput.* **7**(3), 205–230 (1999)
11. 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)
12. Dong, C., Zeng, B.: Inferring parameters through inverse multiobjective optimization. [arXiv:1808.00935](https://arxiv.org/abs/1808.00935) (2018)
13. Eckstein, J., Peitz, S., Schäfer, K., Friedel, P., Köhler, U., Hessel-von Molo, M., Ober-Blöbaum, S., Dellnitz, M.: A comparison of two predictive approaches to control the longitudinal dynamics of electric vehicles. *Procedia Technol.* **26**, 465–472 (2016)
14. Ehrgott, M.: *Multicriteria Optimization*. Springer, Berlin (2005)
15. Fliege, J., Xu, H.: Stochastic multiobjective optimization: sample average approximation and applications. *J. Optim. Theory Appl.* **151**(1), 135–162 (2011)

16. Gander, W., Gander, M., Kwok, F.: *Scientific Computing: An Introduction Using Maple and MATLAB*. Springer, Berlin (2014)
17. Gebken, B., Peitz, S., Dellnitz, M.: On the hierarchical structure of Pareto critical sets. *J. Global Optim.* **73**(4), 891–913 (2019)
18. Hartikainen, M.E., Lovison, A.: PAINT–SiCon: constructing consistent parametric representations of Pareto sets in nonconvex multiobjective optimization. *J. Global Optim.* **62**(2), 243–261 (2015)
19. Heuberger, C.: Inverse combinatorial optimization: a survey on problems, methods, and results. *J. Combin. Optim.* **8**(3), 329–361 (2004)
20. Hillermeier, C.: *Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach*. Birkhäuser, Basel (2001)
21. Huang, S., Liu, Z.: On the inverse problem of linear programming and its application to minimum weight perfect k-matching. *Eur. J. Oper. Res.* **112**(2), 421–426 (1999)
22. Kerschke, P., Wang, H., Preuss, M., Grimme, C., Deutz, A., Trautmann, H., Emmerich, M.: Towards analyzing multimodality of continuous multiobjective landscapes. In: Julia, H., Emma, H., Peter, R. L., Manuel, L.-I., Gabriela, O., Ben, P. (eds.) *Parallel Problem Solving from Nature – PPSN XIV*, pp. 962–972. Springer International Publishing, Cham (2016)
23. Keshavarz, A., Wang, Y., Boyd, S.: Imputing a convex objective function. In: *2011 IEEE International Symposium on Intelligent Control*, pp. 613–619 (2011)
24. Kuhn, H.W., Tucker, A.W.: Nonlinear programming. In: *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pp. 481–492. University of California Press (1951)
25. Lotov, A.V., Kamenev, G.K., Berezkin, V.E., Miettinen, K.: Optimal control of cooling process in continuous casting of steel using a visualization-based multi-criteria approach. *Appl. Math. Model.* **29**, 653–672 (2005)
26. Lovison, A., Pecci, F.: Hierarchical stratification of Pareto sets. [arXiv:1407.1755](https://arxiv.org/abs/1407.1755) (2014)
27. Miettinen, K.: *Nonlinear Multiobjective Optimization*. Springer, New York (1998)
28. Otto, S.E., Rowley, C.W.: Linearly recurrent autoencoder networks for learning dynamics. *SIAM J. Appl. Dyn. Syst.* **18**(1), 558–593 (2019)
29. Peitz, S., Ober-Blöbaum, S., Dellnitz, M.: Multiobjective optimal control methods for the Navier–Stokes equations using reduced order modeling. *Acta Applicandae Mathematicae* **161**(1), 171–199 (2019)
30. Peitz, S., Schäfer, K., Ober-Blöbaum, S., Eckstein, J., Köhler, U., Dellnitz, M.: A multiobjective MPC approach for autonomously driven electric vehicles. *IFAC PapersOnLine* **50**(1), 8674–8679 (2017)
31. Rudin, W.: *Functional Analysis*, 2nd edn. McGraw-Hill, New York (1991)
32. Schaeffer, S.E.: Graph clustering. *Comput. Sci. Rev.* **1**(1), 27–64 (2007)
33. Schilders, W., van der Vorst, H.A., Rommes, J. (eds.): *Model Order Reduction*. Springer, Berlin (2008)
34. Schütze, O., Dell’Aere, A., Dellnitz, M.: On continuation methods for the numerical treatment of multi-objective optimization problems. In: J. Branke, K. Deb, K. Miettinen, R.E. Steuer (eds.) *Practical Approaches to Multi-Objective Optimization*, no. 04461 in *Dagstuhl Seminar Proceedings*. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, Dagstuhl, Germany (2005)
35. Tibshirani, R.: Regression shrinkage and selection via the Lasso. *J. Roy. Stat. Soc.: Ser. B (Methodol.)* **58**(1), 267–288 (1996)
36. Voutchkov, I., Keane, A.: *Multi-Objective Optimization Using Surrogates*, pp. 155–175. Springer, Berlin (2010)
37. Zhang, Q., Zhou, A., Zhao, S., Suganthan, P., Liu, W., Tiwari, S.: *Multiobjective optimization Test Instances for the CEC 2009 Special Session and Competition*. Mechanical Engineering (2008)