The Directed Search Method for Unconstrained Parameter Dependent Multi-objective Optimization Problems

Víctor Adrián Sosa Hernández, Adriana Lara, Heike Trautmann, Günter Rudolph and Oliver Schütze

Abstract In this chapter we present the adaptions of the recently proposed Directed Search method to the context of unconstrained parameter dependent multi-objective optimization problems (PMOPs). The new method, called λ-DS, is capable of performing a movement both toward and along the solution set of a given differentiable PMOP. We first discuss the basic variants of the method that use gradient information and describe subsequently modifications that allow for a gradient free realization. Finally, we show that λ-DS can be used to understand the behavior of stochastic local search within PMOPs to a certain extent which might be interesting for the development of future local search engines, or evolutionary strategies, for the treatment of such problems. We underline all our statements with several numerical results indicating the strength of the novel approach.

Keywords Parameter dependent multi-objective optimization · Local search · Descent method · Continuation method · Stochastic local search · Evolutionary algorithms

V.A. Sosa Hernández (✉) · O. Schütze
Computer Science Department, CINVESTAV-IPN, Av. IPN 2508,
Col. San Pedro Zacatenco, 07360 Mexico City, Mexico
e-mail: msosa@computacion.cs.cinvestav.mx

O. Schütze
e-mail: schuetze@cs.cinvestav.mx

A. Lara
Escuela Superior de Física y Matemáticas del IPN, Mexico City, Mexico
e-mail: adriana@esfm.ipn.mx

H. Trautmann
Department of Information Systems, University of Münster, Leonardo-Campus 3,
48149 Munster, Germany
e-mail: trautmann@uni-muenster.de

G. Rudolph
Fakultät Für Informatik, Technische Universität Dortmund, 44221 Dortmund, Germany
e-mail: Guenter.Rudolph@tu-dortmund.de

O. Schütze et al. (eds.), *NEO 2015*, Studies in Computational
Intelligence 663, DOI 10.1007/978-3-319-44003-3_12

1 Introduction

In many real-world applications one is faced with the problem that several objectives have to be optimized concurrently. Such problems are termed multi-objective optimization problems (MOPs) in literature. One important characteristic of a MOP is that its solution set, the Pareto set, respectively its image, the Pareto front, typically form $(k-1)$-dimensional objects, where k is the number of objectives involved in the problem. Furthermore, it can happen that the MOP contains one or several external parameters $\lambda \in \Lambda \subset \mathbb{R}^l$ such as the temperature of a given mechanical system or the side-wind that impacts a vehicle to be optimized. Such parameters can certainly not be 'optimized' (i.e., they can not be treated as 'normal' decision variables), but can —on the other hand— also not be neglected by the decision maker of the problem. Since for every fixed value of $\lambda \in \Lambda$ such a problem can be viewed as a 'classical' MOP, the solution set of such a *parameter dependent multi-objective optimization problem* (PMOP) is given by an entire family $P_{Q,\Lambda}$ of Pareto sets. Hence, we can expect that the solution set of a given PMOP is of dimension $k-1+l$. Questions that arise in the numerical treatment of PMOPs are for instance to compute a finite size representation of the entire set $P_{Q,\Lambda}$ which has been particularly addressed using specialized evolutionary algorithms (e.g., [4, 5, 7, 13, 37, 42]). Another interesting question is to compute a path of solutions along $P_{Q,\Lambda}$ which is for instance done in the context of robust optimization [11, 40, 41].

In this work, we adapt and extend the Directed Search (DS) method that has recently been proposed in [34] for the treatment of MOPs to the context of PMOPs. The new method, λ-DS, is as DS a point-wise iterative local search method capable of steering the search into any direction d given in objective space. Based on this insight, we will in this chapter analyze the possibility of λ-DS to steer the search toward and along the solution set leading to a new hill climber and a new continuation method. The straightforward implementations of the algorithms require gradient information, however, they can also be realized gradient free if existing neighborhood information is exploited. This makes the λ-DS an ideal candidate for a local search engine within set based search algorithms such as specialized evolutionary algorithms as in these cases the information given by the populations can be exploited. Thus, the methods can ideally be applied 'for free', i.e., without spending additional function evaluations. Finally, we show that the approach of the λ-DS can also be used to explain to a certain extent the behavior of stochastic local search (SLS) within PMOPs. SLS is an important aspect of every stochastic search algorithm such as e.g. evolutionary algorithms. It can be shown that SLS is well-conditioned, i.e., that a pressure both toward and along the solution set are already inherent in SLS. We conjecture that these insights will have a positive influence on the design of future local search mechanisms such as mutation operators or entire stochastic search methods such as specialized evolutionary algorithms.

The remainder of this chapter is organized as follows: in Sect. 2, we briefly state the required background for MOPs and PMOPs and present the related work for the latter. In Sect. 3, we present λ-DS, the Directed Search method for PMOPs and discuss

a descent and a continuation method. In Sect. 4, we will discuss a particular way to realize λ-DS without using the objectives gradients' but via utilizing neighborhood information. In Sect. 5, we consider some aspects of stochastic local search within PMOPs which we underline by some computations. Finally, we draw our conclusions and give paths for possible future research in Sect. 6.

2 Background and Related Work

Here, we briefly state some basic facts on MOPs and PMOPs required for the understanding of this chapter, and state the related work for PMOPs. For a more thorough discussion we refer e.g. to [12, 24]. The most frequently used symbols of this work can be found in Table 1.

2.1 Multi-objective Optimization

A continuous multi-objective optimization problem (MOP) can be stated as follows:

$$\min_{x \in Q} F(x), \tag{1}$$

where F is defined as the vector of the objective functions

Table 1 Most frequently used symbols used in this chapter

F, F_λ	Objective mapping
x	Decision variable
λ	External parameter
Q	Domain in decision variable space
Λ	Domain in parameter space
P_Q	Pareto set
$F(P_Q)$	Pareto front
$P_{Q,\Lambda}$	Family of Pareto sets
$F(P_{Q,\Lambda})$	Family of Pareto fronts
$\nabla f(x)$	Gradient of f at x
$J(x)$	Jacobian of F at x
A^+	Pseudo inverse of matrix A
v	Search direction in decision variable space
d	Direction in objective space
$\eta(x)$	Tangent vector of $P_{Q,\lambda}$ at x

$$\begin{aligned} F : Q &\to \mathbb{R}^k, \\ F(x) &= (f_1(x), \ldots, f_k(x))^T, \end{aligned} \tag{2}$$

and where each objective is given by $f_i : Q \to \mathbb{R}$. We will assume that all the objectives are continuously differentiable except for the extension in Sect. 4. In this study we will focus on unconstrained problems, i.e., problems of the form (1) where the domain is given by $Q = \mathbb{R}^n$.

The optimality of a MOP is defined by the concept of *dominance* [26].

Definition 1 (a) Let $v, w \in \mathbb{R}^k$. Then the vector v is *less than* w ($v <_p w$), if $v_i < w_i$ for all $i \in \{1, \ldots, k\}$. The relation $\leq_p$ is defined analogously.

(b) A vector $y \in Q$ is *dominated* by a vector $x \in Q$ ($x \prec y$) with respect to (1) if $F(x) \leq_p F(y)$ and $F(x) \neq F(y)$, else y is called non-dominated by x.

(c) A point $x \in Q$ is called *(Pareto) optimal* or a *Pareto point* if there is no $y \in Q$ which dominates x.

(d) The set of all Pareto optimal solutions

$$P_Q := \{x \in Q, \text{ s.t. } x \text{ is a Pareto point of } (1)\}, \tag{3}$$

is called the *Pareto set* and its image $F(P_Q)$ the *Pareto front*.

Both, Pareto set and front form typically—i.e., under certain mild regularity assumptions on the objectives—a $(k-1)$-dimensional object (see [17]) which gives rise to multi-objective continuation methods (e.g., [10, 17, 20–22, 27–29]).

If the objectives of the given problem are differentiable one can state a necessary condition for (local) optimality analog to the scalar objective case. Here, we state the theorem for unconstrained problems. For further variants the reader is referred e.g. to [19, 24].

Theorem 1 *Let x^* be a Pareto point of (1), then there exists a vector $\alpha \in \mathbb{R}^k$ with $\alpha_i \geq 0$, $i = 1, \ldots, k$, and $\sum_{i=1}^k \alpha_i = 1$ such that*

$$\sum_{i=1}^{k} \alpha_i \nabla f_i(\mathbf{x}^*) = J(x^*)^T \alpha = 0. \tag{4}$$

Hereby, $J(x)$ denotes the Jacobian of F at x,

$$J(x) = \begin{pmatrix} \nabla f_1(\mathbf{x})^T \\ \vdots \\ \nabla f_k(\mathbf{x})^T \end{pmatrix} \in \mathbb{R}^{k \times n}. \tag{5}$$

A vector $\alpha \in \mathbb{R}^k$ is called a convex weight if $\alpha_i \geq 0$, $i = 1, \ldots, k$, and $\sum_{i=1}^k \alpha_i = 1$. Equation (4) says that the zero vector can be expressed as a convex combination

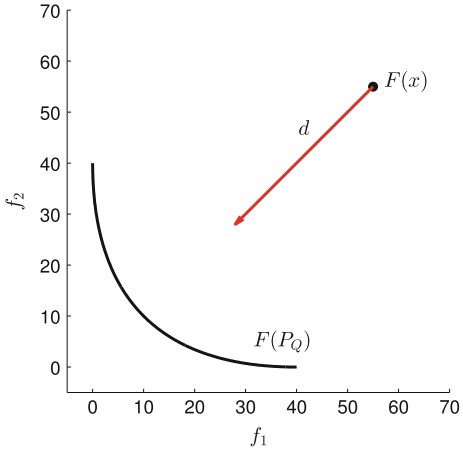


Fig. 1 Idea of the DS: to steer the search into a given direction d in objective space

of the gradients $\nabla f_i(\mathbf{x}^*)$ at each Pareto point $\mathbf{x}^*$. A point $\mathbf{x} \in Q$ satisfying Eq. (4) is called a *Karush–Kuhn–Tucker point*[1] or short a KKT point.

Proof See [18]. □

Recently, the Directed Search (DS) Method has been proposed that allows to steer the search from a given point into a desired direction $d \in \mathbb{R}^k$ in objective space ([34], see also Fig. 1). To be more precise, given a point $x \in \mathbb{R}^n$, a search direction $v \in \mathbb{R}^n$ is sought such that

$$\lim_{t \searrow 0} \frac{f_i(x_0 + tv) - f_i(x_0)}{t} = d_i, \quad i = 1, \ldots, k. \tag{6}$$

Such a direction vector v solves the following system of linear equations:

$$J(x_0)v = d. \tag{7}$$

Since typically $k \ll n$, we can assume that the system in Eq. (7) is (highly) underdetermined. Among the solutions of Eq. (7), the one with the least 2-norm can be viewed as the greedy direction for the given context. This solution is given by

$$v_+ := J(x)^+ d, \tag{8}$$

where $J(x)^+$ denotes the pseudo inverse of $J(x)$. Searching along the direction d is equivalent to find the numerical solution of the following initial value problem (IVP), starting from solution $x_0 \in \mathbb{R}^n$:

[1] Named after the works of Karush [18] and Kuhn and Tucker [19].

$$\begin{aligned} x(0) &= x_0 \in \mathbb{R}^n, \\ \dot{x}(t) &= v_+(x(t)), \quad t > 0, \end{aligned} \tag{9}$$

where $t \in \mathbb{R}$ denotes the time. Since there is in principle no restriction on d the search can be steered in any direction, e.g., toward and along the Pareto set. If d is a 'descent direction' (i.e., $d_i \leq 0$ for all $i = 1, \ldots, k$ and there exists an index j such that $d_j < 0$), a numerical solution of (9) can be viewed as a particular hill climber for MOPs which shares many characteristics with the one described in [3].

In order to measure the approximation quality of a given candidate set to the Pareto set/front we will use the averaged Hausdorff distance, which is particularly used in this context as the following defined Δ_p indicator:

Definition 2 ([32]) Let $A, B \subset \mathbb{R}^n$ be finite sets. The value

$$\Delta_p(A, B) = \max(GD_p(A, B), IGD_p(A, B)), \tag{10}$$

where A is the approximation finite archive, B is the reference finite archive,

$$GD_p(A, B) = \left(\frac{1}{|A|} \sum_{a \in A} d(a, B)^p \right)^{1/p}, \; IGD_p(A, B) = \left(\frac{1}{|B|} \sum_{b \in B} d(b, A)^p \right)^{1/p}, \tag{11}$$

and

$$d(a, B) = \inf_{b \in B} ||a - b||_2, \tag{12}$$

and $p \in \mathbb{N}$, is called the *averaged Hausdorff distance* between A and B.

The indicator Δ_p can be viewed as a composition of slight variations of the Generational Distance (GD, see [38]) and the Inverted Generational Distance (IGD, see [8]). It averages the Hausdorff distance (d_H, see [15]) for finite values of p and coincides with d_H for $p = \infty$. It is worth to notice that, as opposed to that distance, Δ_p does for low values of p in particular not punish single (or few) outliers in a candidate set. On the other hand, for $p < \infty$, Δ_p is only a semi-distance as the triangle inequality does not hold any more. We have chosen this indicator as it, roughly speaking, prefers evenly spread solutions along the Pareto front (e.g., [30]).

2.2 Parameter Dependent Multi-objective Optimization

A continuous parameter dependent multi-objective optimization problem (PMOP) can be stated as follows:

$$\min_{\mathbf{x} \in Q} F_\lambda(x). \tag{13}$$

Hereby, F_λ is defined as a vector of objective functions

$$\begin{aligned} &F_\lambda : Q \to \mathbb{R}^k, \\ &F_\lambda(x) = (f_{1,\lambda}(x), \ldots, f_{k,\lambda}(x))^T, \end{aligned} \tag{14}$$

where $Q \subset \mathbb{R}^n$ is the domain (here we will also consider unconstrained problems, i.e., $Q = \mathbb{R}^n$) and $\lambda \in \Lambda \subset \mathbb{R}^l$ specifies the external parameters to the objective functions. For the domain Λ in the external parameter space we will mainly discuss the unconstrained case (i.e., $\Lambda = \mathbb{R}^l$), but the models we consider here will also have simple bound constraints, i.e.,

$$m_i \leq \lambda_i \leq M_i, \quad i = 1, \ldots, l, \tag{15}$$

where m_i and M_i, $i = 1, \ldots, l$, are lower and upper bounds, respectively. We assume that all objectives $f_{i,\lambda}$, $i = 1, \ldots, k$, are continuously differentiable which we will relax in Sect. 4.

Note that for every fixed value of λ problem (13) can be seen as a classical MOP. Thus, the solution set of (13) consists of an entire family of Pareto sets which is defined as follows:

$$P_{Q,\Lambda} := \{(x, \lambda) \in \mathbb{R}^{n+l}, \text{ such that } x \text{ is a Pareto point of (13) for some } \lambda \in \Lambda\}. \tag{16}$$

The respective family of Pareto fronts is denoted by $F(P_{Q,\Lambda})$. According to the discussion on MOPs we can thus expect that both $P_{Q,\Lambda}$ and $F(P_{Q,\Lambda})$ form under certain (mild) regularity assumptions $(k - 1 + l)$-dimensional objects.

As general examples we consider in this chapter (among other problems) the following three PMOPs that yield different characteristics. The first one is proposed in [31]:

$$\begin{aligned} F_\lambda &: \mathbb{R}^2 \to \mathbb{R}^2 \\ F_\lambda(x) &:= (1 - \lambda)F_1(\mathbf{x}) + \lambda F_2(\mathbf{x}), \end{aligned} \tag{17}$$

where $\lambda \in [0, 1]$ and $F_1, F_2 : \mathbb{R}^2 \to \mathbb{R}^2$,

$$\begin{aligned} F_1(x_1, x_2) &= \begin{pmatrix} (x_1 - 1)^4 + (x_2 - 1)^2 \\ (x_1 + 1)^2 + (x_2 + 1)^2 \end{pmatrix}, \\ F_2(x_1, x_2) &= \begin{pmatrix} (x_1 - 1)^2 + (x_2 - 1)^2 \\ (x_1 + 1)^2 + (x_2 + 1)^2 \end{pmatrix}. \end{aligned}$$

This problem, called PMOP1 in the sequel, is a convex homotopy of the MOPs F_1 and F_2 which have both convex Pareto fronts. Figure 2 shows the sets $P_{Q,\Lambda}$ and $F(P_{Q,\Lambda})$ of PMOP (17).

The second problem, PMOP2, is defined as follows:

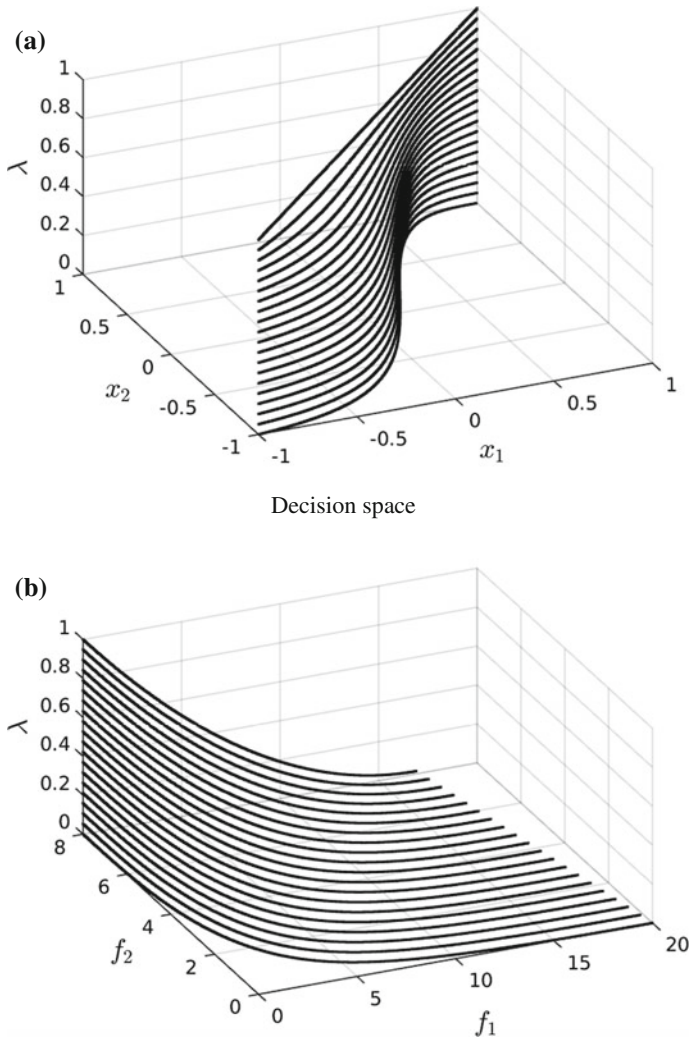


Fig. 2 Family of Pareto sets and their corresponding fronts for PMOP1 (see Eq. (17))

$$
\begin{aligned}
&f_1, f_2 : \mathbb{R}^3 \to \mathbb{R} \\
&f_1(x, \lambda) = \frac{1}{2}\left(\sqrt{1 + (x_1 + x_2)^2} + \sqrt{1 + (x_1 - x_2)^2} + x_1 - x_2\right) + \lambda \cdot e^{-(x_1 - x_2)^2}, \\
&f_2(x, \lambda) = \frac{1}{2}\left(\sqrt{1 + (x_1 + x_2)^2} + \sqrt{1 + (x_1 - x_2)^2} - x_1 + x_2\right) + \lambda \cdot e^{-(x_1 - x_2)^2},
\end{aligned}
\tag{18}
$$

where $\lambda \in [0, 3]$. The objective functions f_i, $i = 1, 2$, are taken from [39]. The Pareto fronts of PMOP2 are convex for $\lambda \in [0, 0.5)$, linear for $\lambda = 0.5$, convex-concave for

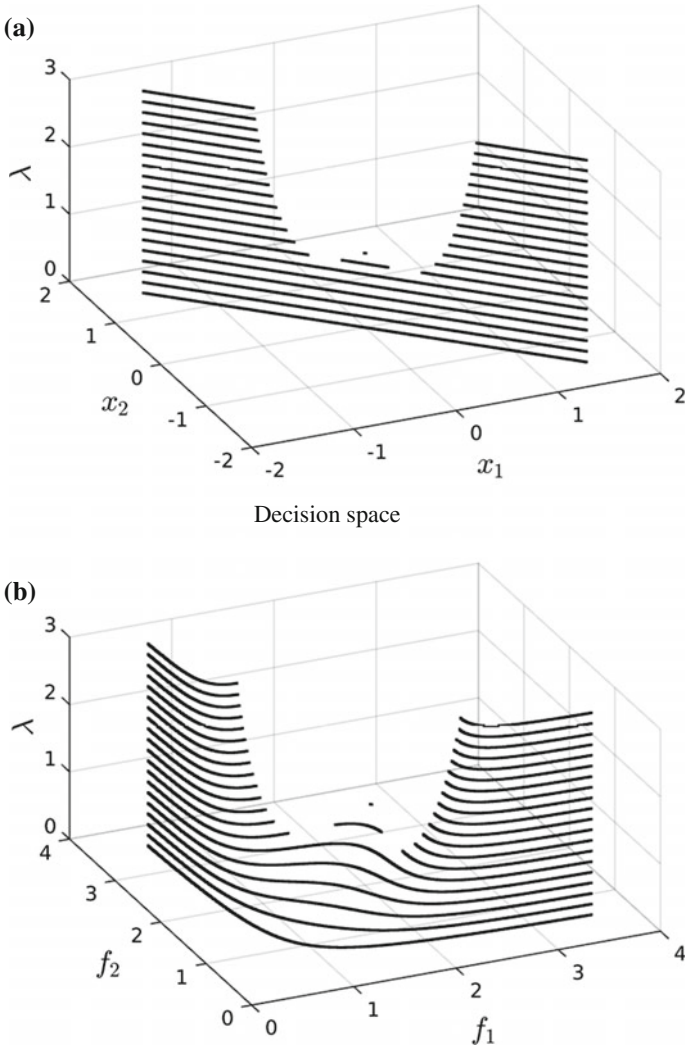


Fig. 3 Family of Pareto sets and their corresponding fronts for PMOP2 (see Eq. (18))

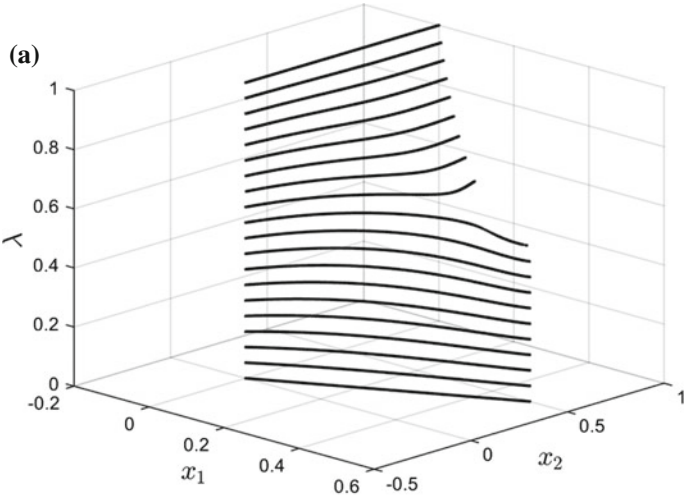
$\lambda \in (0.5, 1.5]$ and disconnected for $\lambda \in (1.5, 3]$. Figure 3 shows the sets $P_{Q,\Lambda}$ and $F(P_{Q,\Lambda})$.

Finally, the third problem, PMOP3, is again a convex homotopy of two MOPs. The first MOP has a convex Pareto front while the other one has a concave front:

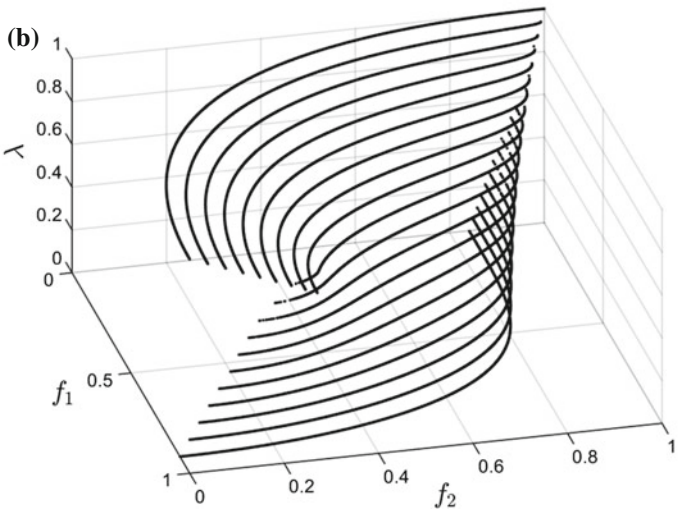
$$\begin{aligned} F_\lambda &: \mathbb{R}^2 \to \mathbb{R}^2 \\ F_\lambda(x) &:= (1-\lambda)F_1(x) + \lambda F_2(x), \end{aligned} \tag{19}$$

where $\lambda \in [0, 1]$, $a_1 = 0$, $a_2 = 1$ and $F_1, F_2 : \mathbb{R}^2 \to \mathbb{R}^2$,

$$F_1(\mathbf{x}) = \begin{pmatrix} (x_1^2 + x_2^2)^{0.125} \\ ((x_1 - 0.5)^2 + (x_2 - 0.5)^2)^{0.25} \end{pmatrix},$$
$$F_2(\mathbf{x}) = \begin{pmatrix} x_1^2 + x_2^2 \\ (x_1 - a_1)^2 + (x_2 - a_2)^2 \end{pmatrix}.$$



Decision space



Objective space

Fig. 4 Family of Pareto sets and their corresponding fronts for PMOP3 (see Eq. (19))

The Pareto fronts of PMOP3 are concave for $\lambda \in [0, 0.5]$ and convex for $\lambda \in (0.5, 1]$. Figure 4 shows a family of the sets $P_{Q,\Lambda}$ and $F(P_{Q,\Lambda})$.

2.3 Related Work

Probably the first study in the field of parameter dependent optimization has been published by Manne in 1953 [2]. From this point, researchers have published many important works within evolutionary multi-objective optimization literature.

A classification of dynamic MOPs (which are particular PMOPs where the value of λ changes over the time) is presented in [36]. This work focuses on the components that lead to the observed dynamic behavior. The work of Farina, Deb, and Amato [13] also deals with dynamic MOPs. It contains some test case applications as well as many results related to problems, which depend on an external parameter. Further, a classification of dynamic MOPs is established. The work in [39] gives a good insight into PMOPs, but only treats problems with just one external parameter, by using numerical path following algorithms. Also, some geometrical properties of the solution sets are discussed as well as connections to bifurcation theory. In [4], a survey over the evolutionary techniques that tackle dynamic optimization problems is provided. Four main methods to overcome such problems are described: (i) increasing the diversity after the change of the solution set, (ii) maintaining the diversity over the complete run of the evolutionary algorithm to detect the changes in the solution set, (iii) memory based approaches, and finally, (iv) multi-population approaches, which are the ones that reduce the main problem into subproblems or 'slices' in order to maintain a small population until the family of solution sets is reached.

The idea to use slices, or multi-population approaches, in the evolution of an evolutionary algorithm is used for example in [5]. There, an algorithm is proposed that solves the problem by dividing the objective landscape into sub-populations in order to reach all the solutions over the external parameter (in this case time).

In [43], the basic idea is to use history information in order to numerically solve PMOPs. The authors propose a smart re-initialization of the population in order to take advantage of the previous knowledge leading to two different set based continuation-like methods.

Another approach to solve PMOPs can be found in [35]. In this work the authors use a parallel version of the NSGA-II in order to solve a dynamic optimization problem to reduce the energy consumption. They divide the complete problem into nodes and then the algorithm NSGA-II is executed in each node to compute the solution set. Finally, in [7] the authors present a taxonomy of the ways to treat PMOPs and also mention several similarities and differences between PMOPs and MOPs. Here again the multi-population idea is used and adapted by using migration methods.

Further, there exist also non-evolutionary approaches to PMOPs. In [41], a new concept for online multi-objective optimization is proposed as it is argued that the computation of the entire solution set $P_{Q,\Lambda}$ is too complex in many situations. Instead,

the problem is treated by a novel path following technique that generates solutions along $P_{Q,A}$. This work was later on extended in [11, 40] for the detection of robust Pareto points.

3 The Directed Search Method for PMOPs

Analogously to the DS for the treatment of MOPs, the λ-DS designed for PMOPs can steer the search into any direction that is given in objective space. In this section, we first discuss the basic idea for general directions, and will further on develop a descent strategy and two variants of a continuation method.

3.1 Basic Idea

For our considerations it is advantageous to treat λ—at least formally—within PMOPs as a 'normal' parameter. Thus, we consider in the sequel the following map:

$$\begin{aligned} \tilde{F} &: \mathbb{R}^{n+l} \to \mathbb{R}^{k+l} \\ \tilde{F}(\mathbf{x}, \lambda) &= \begin{pmatrix} \tilde{f}_1(\mathbf{x}, \lambda) \\ \vdots \\ \tilde{f}_{k+l}(\mathbf{x}, \lambda) \end{pmatrix} := \begin{pmatrix} f_1(\mathbf{x}, \lambda) \\ \vdots \\ f_k(\mathbf{x}, \lambda) \\ \lambda \end{pmatrix}, \end{aligned} \tag{20}$$

where $\tilde{f}_i : \mathbb{R}^{n+l} \to \mathbb{R}$, $i = 1, \ldots, k + l$. The Jacobian of $\tilde{F}$ is given by

$$J(\mathbf{x}, \lambda) = \begin{pmatrix} \nabla_x f_1(\mathbf{x}, \lambda)^T & \nabla_\lambda f_1(\mathbf{x}, \lambda)^T \\ \vdots & \vdots \\ \nabla_x f_k(\mathbf{x}, \lambda)^T & \nabla_\lambda f_k(\mathbf{x}, \lambda)^T \\ 0 & I_l \end{pmatrix} := \begin{pmatrix} J_x & J_\lambda \\ 0 & I_l \end{pmatrix} \in \mathbb{R}^{(k+l)\times(n+l)}, \tag{21}$$

where

$$J_x = \begin{pmatrix} \nabla_x f_1(\mathbf{x}, \lambda)^T \\ \vdots \\ \nabla_x f_k(\mathbf{x}, \lambda)^T \end{pmatrix} \in \mathbb{R}^{k\times n}, \quad J_\lambda = \begin{pmatrix} \nabla_\lambda f_1(\mathbf{x}, \lambda)^T \\ \vdots \\ \nabla_\lambda f_k(\mathbf{x}, \lambda)^T \end{pmatrix} \in \mathbb{R}^{k\times l}, \tag{22}$$

and where I_l denotes the $(l \times l)$-identity matrix.

Using (20), we can now formulate the general idea of the Directed Search method for PMOPs, called λ-DS, analog to its variant for MOPs: given a direction

$$d = \begin{pmatrix} d_f \\ d_\lambda \end{pmatrix} \in \mathbb{R}^{k+l} \tag{23}$$

in the compound objective-parameter space, i.e., $d_f \in \mathbb{R}^k$ and $d_\lambda \in \mathbb{R}^l$, a search direction

$$v = \begin{pmatrix} v_f \\ v_\lambda \end{pmatrix} \in \mathbb{R}^{n+l} \tag{24}$$

in the decision space of $\tilde{F}$, where $v_f \in \mathbb{R}^n$ and $v_\lambda \in \mathbb{R}^l$, is sought for such that

$$\lim_{t \to 0} \frac{\tilde{f}_i((\mathbf{x}, \lambda) + tv) - \tilde{f}_i(\mathbf{x}, \lambda)}{t} = \langle \nabla \tilde{f}_i(\mathbf{x}, \lambda), v \rangle = d_i, \ i = 1, \ldots, k+l, \tag{25}$$

where $d = (d_1, \ldots, d_{k+l})^T$. Using the Jacobian of $\tilde{F}$, we can write (25) in matrix-vector notation as

$$\begin{pmatrix} J_x & J_\lambda \\ 0 & I_l \end{pmatrix} \begin{pmatrix} v_f \\ v_\lambda \end{pmatrix} = \begin{pmatrix} d_f \\ d_\lambda \end{pmatrix}. \tag{26}$$

Solving Eq. (26) leads to

$$I_l v_\lambda = d_\lambda \quad \Rightarrow \quad v_\lambda = d_\lambda \tag{27}$$

and

$$J_x v_f + J_\lambda v_\lambda = d_f \quad \Rightarrow \quad J_x v_f = d_f - J_\lambda d_\lambda \tag{28}$$

Equation (27) shows that the search direction v_λ is identical to the desired direction d_λ in parameter space, which follows directly by the construction of $\tilde{F}$. For the movement in decision space, however, λ has a certain influence as Eq. (28) reveals.

The search direction v is computed via solving the linear system of Eq. (26). Note that if $n > k$ and if the rank of $J(x, \lambda)$ is maximal (i.e., $k + l$), then there exists a $(n - k)$-dimensional subspace of vectors that solve (26). Out of them, the solution v_+ with the smallest Euclidean norm can be viewed as the most greedy solution. This vector is given by

$$v_+ = J(x, \lambda)^+ d, \tag{29}$$

where $J(x, \lambda)^+ \in \mathbb{R}^{(n+l)\times(k+l)}$ denotes the pseudo-inverse of $J(x, \lambda)$. From now on, we will use notation F instead of $\tilde{F}$, for simplicity.

The next result states a formula for the computation of the pseudo-inverse of $J(x, \lambda)$ under certain (mild) assumptions.

Proposition 1 *Let*

$$J := \begin{pmatrix} J_x & J_\lambda \\ 0 & I_l \end{pmatrix} \tag{30}$$

be as in Eq. (21) *and assume the following holds:*

1. $rank(J_x) = k$,
2. $J_x J_x^T + J_\lambda J_\lambda^T$ *is invertible, and*
3. $I_l - J_\lambda^T (J_x J_x^T + J_\lambda J_\lambda^T)^{-1} J_\lambda$ *is also invertible.*

Then, the pseudo-inverse of J *is given by*

$$J^+ = \begin{pmatrix} J_x^+ - J_x^+ J_\lambda \\ 0 \quad I_l \end{pmatrix}. \tag{31}$$

Proof Let $J := \begin{pmatrix} J_x & J_\lambda \\ 0 & I_l \end{pmatrix}$. Since the rank of J_x is k the rank of J is $k + l$ (i.e., maximal), and the pseudo inverse[2] of J is given by

$$J^+ = J^T (J J^T)^{-1}. \tag{32}$$

It is

$$J J^T = \begin{pmatrix} J_x & J_\lambda \\ 0 & I_l \end{pmatrix} \begin{pmatrix} J_x^T & 0 \\ J_\lambda^T & I_l \end{pmatrix} = \begin{pmatrix} J_x J_x^T + J_\lambda J_\lambda^T & J_\lambda \\ J_\lambda^T & I_l \end{pmatrix}. \tag{33}$$

We know that the inverse of a block matrix is given by

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} = \begin{pmatrix} S_D^{-1} & -A^{-1} B S_A^{-1} \\ -D^{-1} C S_D^{-1} & S_A^{-1} \end{pmatrix}^{-1}, \tag{34}$$

where

$$S_A = D - C A^{-1} B, \tag{35}$$

$$S_D = A - B D^{-1}. \tag{36}$$

Writing $J J^T$ as in Eq. (34), we obtain

$$A = J_x J_x^T + J_\lambda J_\lambda^T, \tag{37}$$

$$B = J_\lambda, \tag{38}$$

$$C = J_\lambda^T, \tag{39}$$

$$D = I_l. \tag{40}$$

Using Eqs. (37), (38), (39) and (40), we get for S_A and S_D

[2] If the rank of a matrix $A \in \mathbb{R}^{m \times n}$, $m \leq n$, is m (i.e., maximal), its pseudo inverse is given by $A^+ = A^T (A A^T)^{-1} \in \mathbb{R}^{n \times m}$.

$$S_A = I_l - J_\lambda^T (J_x J_x^T + J_\lambda J_\lambda^T)^{-1} J_\lambda, \tag{41}$$

$$S_D = J_x J_x^T + J_\lambda J_\lambda^T - J_\lambda I_l J_\lambda^T = J_x J_x^T + J_\lambda J_\lambda^T - J_\lambda J_\lambda^T = J_x J_x^T. \tag{42}$$

By replacing these values into the blocks of Eq. (34), we get the following:

$$S_A^{-1} = (I_l - J_\lambda^T (J_x J_x^T + J_\lambda J_\lambda^T)^{-1} J_\lambda)^{-1}, \tag{43}$$

$$S_D^{-1} = (J_x J_x^T)^{-1}, \tag{44}$$

$$\begin{aligned}
-A^{-1} B S_A^{-1} &= -((J_x J_x^T + J_\lambda J_\lambda^T)^{-1} J_\lambda (I_l - J_\lambda^T (J_x J_x^T + J_\lambda J_\lambda^T)^{-1} J_\lambda)^{-1}) \\
&= -((I_l - J_\lambda^T (J_x J_x^T + J_\lambda J_\lambda^T)^{-1} J_\lambda) J_\lambda^{-1} (J_x J_x^T + J_\lambda J_\lambda^T))^{-1} \\
&= -(J_\lambda^{-1} (J_x J_x^T + J_\lambda J_\lambda^T) - J_\lambda^T (J_x J_x^T + J_\lambda J_\lambda^T)^{-1} J_\lambda J_\lambda^{-1} (J_x J_x^T + J_\lambda J_\lambda^T))^{-1} \\
&= -(J_\lambda^{-1} (J_x J_x^T + J_\lambda J_\lambda^T) - J_\lambda^T)^{-1} \\
&= -(J_\lambda^{-1} J_x J_x^T + J_\lambda^{-1} J_\lambda J_\lambda^T - J_\lambda^T)^{-1} \\
&= -(J_\lambda^{-1} J_x J_x^T + J_\lambda^T - J_\lambda^T)^{-1} \\
&= -(J_\lambda^{-1} J_x J_x^T)^{-1} \\
&= -((J_x J_x^T)^{-1} J_\lambda),
\end{aligned} \tag{45}$$

and

$$-D^{-1} C S_D^{-1} = -(I_l J_\lambda^T (J_x J_x^T)^{-1}) = -(J_\lambda^T (J_x J_x^T)^{-1}). \tag{46}$$

Therefore,

$$(JJ^T)^{-1} = \begin{pmatrix} (J_x J_x^T)^{-1} & -((J_x J_x^T)^{-1} J_\lambda) \\ -(J_\lambda^T (J_x J_x^T)^{-1}) & (I_l - J_\lambda^T (J_x J_x^T + J_\lambda J_\lambda^T)^{-1} J_\lambda)^{-1} \end{pmatrix}. \tag{47}$$

Using Eq. (32), we obtain:

$$\begin{aligned}
J^+ &= J^T (JJ^T)^{-1} \\
&= \begin{pmatrix} J_x^T & 0 \\ J_\lambda^T & I_l \end{pmatrix} \begin{pmatrix} (J_x J_x^T)^{-1} & -((J_x J_x^T)^{-1} J_\lambda) \\ -(J_\lambda^T (J_x J_x^T)^{-1}) & (I_l - J_\lambda^T (J_x J_x^T + J_\lambda J_\lambda^T)^{-1} J_\lambda)^{-1} \end{pmatrix} \\
&= \begin{pmatrix} J_x^T (J_x J_x^T)^{-1} & -(J_x^T (J_x J_x^T)^{-1} J_\lambda) \\ J_\lambda^T (J_x J_x^T)^{-1} - (J_\lambda^T (J_x J_x^T)^{-1}) & I_l \end{pmatrix}.
\end{aligned} \tag{48}$$

Finally, replacing J_x^+ by $J_x^T (J_x J_x^T)^{-1}$ in Eq. (48) leads to

$$J^+ = \begin{pmatrix} J_x^+ & -J_x^+ J_\lambda \\ 0 & I_l \end{pmatrix}, \tag{49}$$

as claimed. □

The following example illustrates that the choice of ν_+ as it was defined in Eq. (29), is indeed preferable among the solutions ν that solve Eq. (26). For simplicity but without loss of generality, we consider a MOP (i.e., neglecting the external parameter).

Example 1 Consider the bi-objective problem

$$\begin{aligned} f_i :&\mathbb{R}^{10} \to \mathbb{R}, \quad i = 1, 2, \\ f_i(x) &= \|x - a^i\|^2, \quad i = 1, 2, \end{aligned} \tag{50}$$

where $a^1 = (1, \ldots, 1)^T$ and $a^2 = (-1, \ldots, -1)^T$. The Pareto set of problem (50) is the line segment that connects a^1 and a^2. For the (non-optimal) vector $x_0 = (1, -1, \ldots, 1, -1)^T \in \mathbb{R}^{10}$ it is

$$J(x_0) = 4 \begin{pmatrix} 0 & -1 & \cdots & 0 & -1 \\ 1 & 0 & \cdots & 1 & 0 \end{pmatrix} \in \mathbb{R}^{2\times 10}, \quad \text{and} \tag{51}$$

$$J(x_0)^+ = \frac{1}{20} \begin{pmatrix} 0 & 1 \\ -1 & 0 \\ \vdots & \vdots \\ 0 & 1 \\ -1 & 0 \end{pmatrix} \in \mathbb{R}^{10\times 2}. \tag{52}$$

For the direction in objective space $d = (-1, -1)^T$ we obtain

$$\nu_+ = J^+ d = \frac{1}{20}(-1, 1, \ldots, -1, 1)^T. \tag{53}$$

We consider the vectors

$$\begin{aligned} \bar{\nu}_1 &:= \frac{1}{4}(-1, 1, 0, \ldots, 0)^T, \\ \bar{\nu}_2 &:= \frac{1}{8}(-1, 1, -1, 1, 0, \ldots, 0)^T, \\ \bar{\nu}_3 &:= \frac{1}{12}(-1, 1, -1, 1, -1, 1, 0, \ldots, 0)^T, \\ \bar{\nu}_4 &:= \frac{1}{16}(-1, 1, -1, 1, -1, 1, -1, 1, 0, 0)^T, \\ \bar{\nu}_5 &:= \frac{1}{20}(-1.1, \ldots, -1, 1)^T = \nu_+. \end{aligned} \tag{54}$$

The $\bar{\nu}_i$'s are orthogonal projections (and thus best approximations) of ν_+ onto the subspaces

$$S_i := \left\{x \in \mathbb{R}^{10} \ : \ x_{2i+1} = \cdots = x_n = 0\right\}, \quad i = 1, \ldots, 5. \tag{55}$$

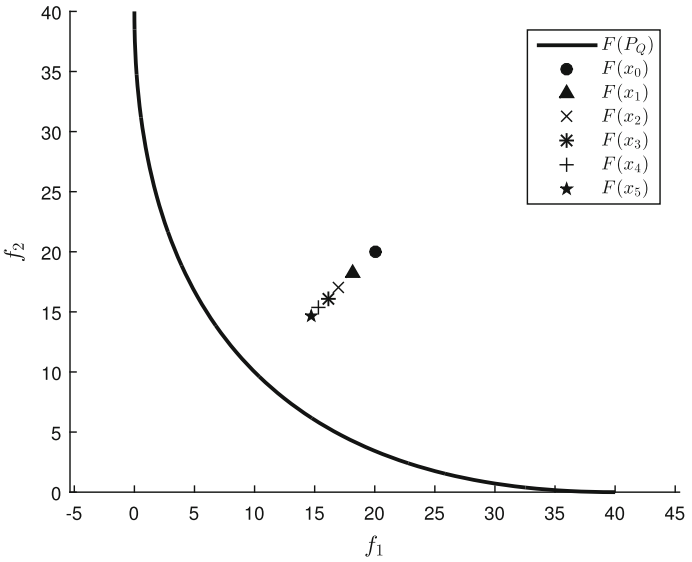


Fig. 5 Generated images $F(x_i)$ for $i = 0, 1 \ldots, 5$ by Eq. (57) for PMOP1 using $x_0 = (1, -1, \ldots, 1, -1)^T \in \mathbb{R}^{10}$ as starting point

Further, it is $S_i \subset S_{i+1}$, $i = 1, \ldots, 4$, and $S_5 = \mathbb{R}^{10}$. For all $\bar{\nu}_i$'s it holds

$$J(x_0)\bar{\nu}_i = d, \quad i = 1, \ldots, 5. \tag{56}$$

Figure 5 shows the Pareto front of MOP (50) as well as the images $F(x_i), i = 0, \ldots, 5$, where

$$x_i := x_0 + \bar{t}\frac{\bar{\nu}_i}{\|\bar{\nu}_i\|}. \tag{57}$$

Hereby, we have fixed $\bar{t}$ to 1 and have normalized the direction vectors $\bar{\nu}_i$ in order to obtain the same step size. As it can be seen, the movement in objective space in direction d is larger the 'closer' the $\bar{\nu}_i$'s come to ν_+. Thus, $\nu_+ = \bar{\nu}_5$ can be considered the best direction, which is indeed in general the case.

3.2 The λ-DS Descent Method

After having stated the idea of the Directed Search method for PMOPs we are now in the position to steer the search into particular directions. First, we are interested in a movement toward $P_{Q,\Lambda}$. For this, we assume we are given a search direction d where $d_f \in \mathbb{R}^k \backslash \{0\}$ with $d_{f,i} \leq 0$, $i = 1, \ldots, k$, while there is no restriction on d_λ. Further, we assume that we are given a starting point $(x_0, \lambda_0) \in \mathbb{R}^n$ with $rank(J_x(x_0)) = k$ and the image for F_λ where F_λ is bounded from below for all $\lambda \in \Lambda$. A greedy search

in d-direction leads thus to the (numerical) solution of the following IVP:

$$\begin{aligned} z(0) &= (x_0, \lambda_0) \in \mathbb{R}^{n+l} \\ \dot{z}(t) &= J(x(t), \lambda(t))^+ d. \end{aligned} \tag{58}$$

We will now investigate solutions of (58) qualitatively. Let $\gamma : [0, t_f] \to \mathbb{R}^{n+l}$ be such a solution, and let t_c be the smallest value of $t \geq 0$ such that

$$\nexists v \in \mathbb{R}^n : \quad J(x(t), \lambda(t))v = d. \tag{59}$$

We will call t_c the critical value and $\gamma(t_c)$ the critical point of (58). γ can be divided into two parts: into $\gamma([0, t_c])$ and $\gamma([t_c, t_f])$. In the first part, $F(\gamma(t))$ yields the desired decay in d-direction. The critical point $\gamma(t_c)$ may be a boundary point. From there on, a 'best fit' for the tangent is computed (which follows directly by the properties of the pseudo-inverse), i.e.,

$$v_+(x(t)) := \arg\min_{v \in \mathbb{R}^n} \|J(x(t), \lambda(t))v - d\| \tag{60}$$

is chosen. For the end point $\gamma(t_f)$ it holds

$$J(\gamma(t_f))^+ d = 0. \tag{61}$$

Such end points are on the one hand certainly interesting as they are KKT points of $F_{\lambda(t_f)}$, with associated weight

$$\alpha = -\frac{d}{\|d\|_1}. \tag{62}$$

To see this, let

$$J(\gamma(t_f), \lambda(t_f)) = U \Sigma V^T \tag{63}$$

be a singular value decomposition of $J(\gamma(t_f))$. Since $J(\gamma(t_f))^+ d = 0$, where $J(\gamma(t_f))^+ = V \Sigma^+ U^T$, it is also $J(\gamma(t_f))^T d = V \Sigma U^T d = 0$. That is, it holds

$$J(\gamma(t_f))^T d = \begin{pmatrix} J_x^T & 0 \\ J_\lambda^T & I_l \end{pmatrix} \begin{pmatrix} d_f \\ d_\lambda \end{pmatrix} = 0. \tag{64}$$

From the first equation in (64) we see that

$$J_x^T d_f = 0 \tag{65}$$

and thus also

$$\sum_{i=1}^{k} \alpha_i \nabla f_{i,\lambda(t_f)}(x(t_f)) = 0. \tag{66}$$

That is, $x(t_f)$ is indeed a KKT point of $F_{\lambda(t_f)}$.

On the other hand, the computation of γ in $[t_c, t_f]$ might get computationally expensive since Eq. (58) is stiff in the second part as the rank of J_x is less than k and thus the rank of J is less than $k + l$ (i.e., not maximal). Furthermore, the computation of $\gamma([t_c, t_f])$ does not fit with the original idea of the directed search. Hence, we will restrict ourselves in the sequel to the detection of the critical points.

As seen above, one can not expect to get KKT points when computing the critical points. The following result, however, shows a certain relation to the Normal Boundary Intersection (NBI, see [9]) method that is widely used for the treatment of MOPs. The NBI-subproblem for a given point (x_0, λ_0) adapted to the context of PMOPs reads as follows[3]:

$$\begin{aligned} &\max_{x,\lambda,t} \; t \\ &\text{s.t. } \; F(x, \lambda) = F(x_0, \lambda_0) + td. \end{aligned} \tag{67}$$

Proposition 2 *Let (x^*, λ^*) be a critical point of IVP (58), then it is a local solution of problem (67).*

Proof Let $g(x, \lambda, t) := t$ and $h_i(x, \lambda, t) := f_i(x_0, \lambda_0) + td_i - f_i(x, \lambda)$, $i = 1, \ldots, k + l$. Assume that (x^*, λ^*) is not a local solution of Eq. (67). Then, there exist vectors $\nu = (\tilde{\nu_f}, \tilde{\nu_\lambda}, \nu_{n+l+1}) \in \mathbb{R}^{n+l+1}$, $\tilde{\nu_f} \in \mathbb{R}^n$, $\tilde{\nu_\lambda} \in \mathbb{R}^l$ and a scalar $t^* \in \mathbb{R}$ such that

$$\langle \nabla g(\mathbf{x}^*, \lambda^*, t^*), \nu \rangle = \left\langle \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} \tilde{\nu_f} \\ \tilde{\nu_\lambda} \\ \nu_{n+l+1} \end{pmatrix} \right\rangle > 0, \text{ and} \tag{68}$$

$$\langle \nabla h_i(\mathbf{x}^*, \lambda^*, t^*), \nu \rangle = \left\langle \begin{pmatrix} -\nabla_x f_i(\mathbf{x}^*, \lambda^*) \\ -\nabla_\lambda f_i(\mathbf{x}^*, \lambda^*) \\ d_i \end{pmatrix}, \begin{pmatrix} \tilde{\nu_f} \\ \tilde{\nu_\lambda} \\ \nu_{n+l+1} \end{pmatrix} \right\rangle = 0, \, i = 1, \ldots, k + l. \tag{69}$$

By Eq. (68), it follows that $\nu_{n+l+1} \neq 0$, and by Eq. (69), we have that

$$\langle \nabla f_i(\mathbf{x}^*, \lambda^*), \tilde{\nu} \rangle = \nu_{n+l+1} d_i, \; i = 1, \ldots, k, \tag{70}$$

where $\tilde{\nu} = (\tilde{\nu_f}, \tilde{\nu_\lambda})^T$.

Hence, for $\tilde{\nu} := \frac{1}{\nu_{n+l+1}} \tilde{\nu}$ it is $J(\mathbf{x}^*, \lambda^*)\tilde{\nu} = d$ which contradicts that $(\mathbf{x}^*, \lambda^*)$ is a critical point of Eq. (58). □

Remark 1 On the other hand, local solutions of problem (67) are also potential critical points of Eq. (58): let $\mathbf{x}^{**}$ be a solution of problem (67) and assume that there

[3] Furthermore, we note that the original idea of NBI for a given MOP is not to maximize the distance from $F(x_0)$ for a given point x_0, but this is a straightforward adaption to the current context to steer the search in a given direction d.

exists a $\nu \in \mathbb{R}^{n+l}$ such that $J(\mathbf{x}^{**}, \lambda^{**})\nu = d$. Then, $\tilde{\nu} = (\nu, -1) \in \mathbb{R}^{n+l+1}$ satisfies Eqs. (68) and (69) which is in contradiction to the assumption of $(\mathbf{x}^{**}, \lambda^{**})$.

Numerical Realization. For the numerical realization of the λ-DS Descent Method we refer to the corresponding realization of the original DS Descent method [23, 34]. We do this as the description is rather lengthy and since most of its elements can be taken without any or just few adaptions to the problem at hand. An exception is the stopping criterion. Equation (59) shows that the rank of J_x at a critical point is not maximal while the rank is maximal (by assumption) at the starting point x_0. While the rank of a matrix can of course not be used to detect $(x(t_c), \lambda(t_c))$ numerically, we can use the condition number κ_2 of $J_x(x)$

$$\kappa_2(J_x(x, \lambda)) = \|J_x(x, \lambda)\|_2 \|J_x(x, \lambda)^+\|_2 = \frac{\sigma_1}{\sigma_k}, \tag{71}$$

where σ_1 and σ_k are the largest and smallest singular values of $J_x(x)$, respectively. The search is stopped if

$$\kappa_2(J_x(x_i, \lambda_i)) \geq tol, \tag{72}$$

where $tol \in \mathbb{R}_+$ is a given (large) threshold. This can be done since by the above discussion

$$\kappa_2(J_x(x(t), \lambda(t))) \to \infty \quad \text{for} \quad (x(t), \lambda(t)) \to (x(t_c), \lambda(t_c)). \tag{73}$$

The selection of more sophisticated stopping conditions is subject of ongoing research.

Example 2 In the following, we show graphically the behavior of the λ-DS descent method over the three PMOPs defined above. Figure 6 shows the movement produced by λ-DS descent method over PMOP1. The non-optimal vector selected for this example is $(x_0, \lambda_0) = (2.0, -1.5, 0.5)^T$. In the case of PMOP2 and PMOP3, Figs. 7 and 8 show the movement toward the solution set. The used starting points are $(x_0, \lambda_0) = (2.0, 2.0, 0.5)^T$ and $(x_0, \lambda_0) = (-2.0, 0.5, 0.9)^T$, respectively. For each example, we used an Armijo step size control with initial step size $t_0 = 0.5$. As stopping criterion we used $cond(J(x, \lambda)) \geq 100$ for candidate solutions (x, λ). Table 2 contains the number of function and Jacobian evaluations used for computing our method, it also has the number of iterations and the condition number at the final points. Note that the condition number at the final points are in certain cases much larger than the threshold, this is due to the choice of the step size strategy.

3.3 The λ-DS Continuation Method

In this section, we first discuss how to properly choose the predictor direction and present further on a complete predictor–corrector method to perform a movement in

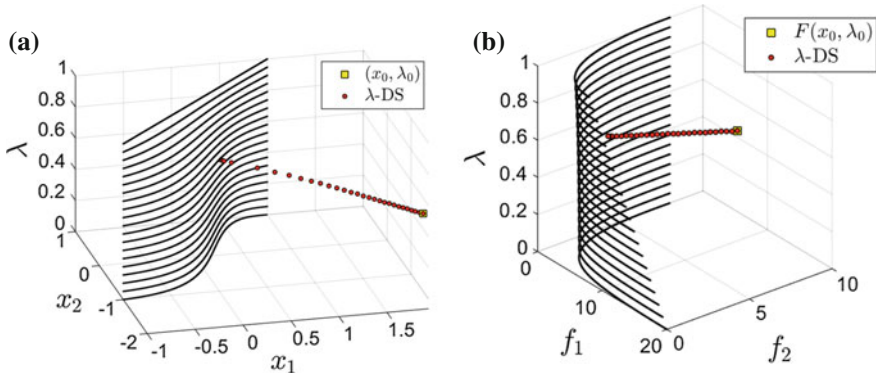


Fig. 6 Result of the λ-DS descent method for PMOP1

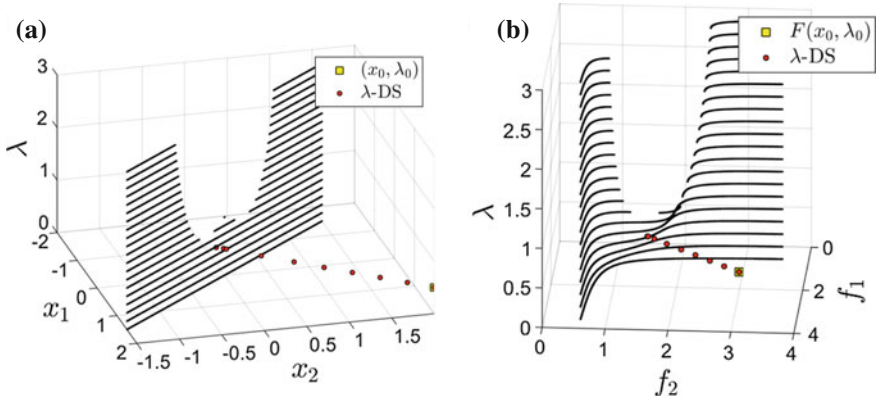


Fig. 7 Result of the λ-DS descent method for PMOP2

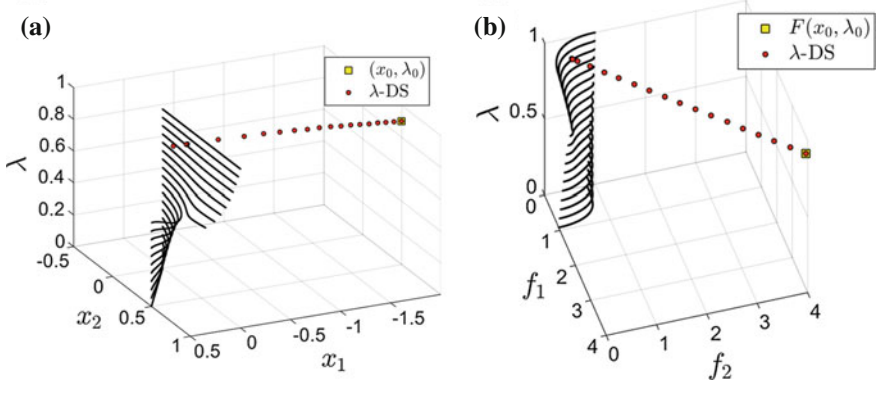


Fig. 8 Result of the λ-DS descent method for PMOP3

Table 2 Function evaluations needed to perform λ-DS descent method

Information	PMOP1	PMOP2	PMOP3
# of $F(\mathbf{x}, \lambda)$	47	22	32
# of $J(\mathbf{x}, \lambda)$	29	10	19
$cond(J(\mathbf{x}, \lambda))$	837.065	131.77	431.965
# of iterations	29	10	19

λ-space within $P_{Q,\Lambda}$. The new continuation method is able to steer the search into any direction along the solution set and is per se applicable without using Hessian information which distincts them from other related algorithms.

Steering the Search for the Predictor. In order to steer the search for the predictor direction we will first have to identify the normal vector η at a given point $F(x, \lambda)$ of $F(P_{Q,\Lambda})$. For this, we have to make some considerations.

Let $z_i := (x_i, \lambda_i) \in \mathbb{R}^{n+l}$ be a given iterate. If the subsequent iterate $z_{i+1} := (x_{i+1}, \lambda_{i+1})$ is chosen via line search, i.e.,

$$z_{i+1} = z_i + t_i v_i, \tag{74}$$

where $t_i \in \mathbb{R}_+$ is a given step size and $v_i \in \mathbb{R}^{n+l}$ a given search direction, then the corresponding movement—for infinitesimal step sizes—in objective space is given by

$$J(z_i)v_i. \tag{75}$$

To see this, consider the j-th component of $J(z_i)v_i$:

$$(J(z_i)v_i)_j = \lim_{t\to 0} \frac{\tilde{f}_j((x_i, \lambda_i) + tv_{i,j}) - \tilde{f}_j(x_i, \lambda_i)}{t} = \langle \nabla \tilde{f}_i(x_i, \lambda_i), v_{i,j} \rangle, \quad j = 1, \ldots, k + l. \tag{76}$$

Further, for MOPs it is known that $J_x v_x$ points along the linearized Pareto front [17, 34]. More precisely, let x be a KKT point of Eq. (1) and $\alpha \in \mathbb{R}^k$ be its associated weight. Then it holds

$$\langle J_x v_x, \alpha \rangle = \langle v_x, J_x^T \alpha \rangle = 0. \tag{77}$$

To derive the normal η for $F(P_{Q,\lambda})$ we proceed as in Eq. (77): assume we are given a KKT point (x, λ) of F_λ with associated weight vector $\alpha \in \mathbb{R}^k$, we are looking for a vector $\eta = (\alpha, \beta) \in \mathbb{R}^{k+l}$ such that for all vectors $v = (v_x, v_\lambda)$ it holds

$$\langle Jv, \eta \rangle = \left\langle \begin{pmatrix} J_x & J_\lambda \\ 0 & I_l \end{pmatrix} \begin{pmatrix} v_f \\ v_\lambda \end{pmatrix}, \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \right\rangle = 0. \tag{78}$$

Solving (78) leads to

$$
\begin{aligned}
0 &= \left\langle \begin{pmatrix} J_x v_f + J_\lambda v_\lambda \\ v_\lambda \end{pmatrix}, \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \right\rangle = \langle J_x v_f + J_\lambda v_\lambda, \alpha \rangle + \langle v_\lambda, \beta \rangle \\
&= \underbrace{\langle J_x v_x, \alpha \rangle}_{=0} + \langle J_\lambda v_\lambda, \alpha \rangle + \langle v_\lambda, \beta \rangle = \langle v_\lambda, {J_\lambda}^T \alpha \rangle + \langle v_\lambda, \beta \rangle \\
&= \langle v_\lambda, {J_\lambda}^T \alpha + \beta \rangle
\end{aligned} \tag{79}
$$

This is satisfied for

$$\beta = -{J_\lambda}^T \alpha. \tag{80}$$

Using this, we obtain for the tangent vector

$$\eta = \begin{pmatrix} \alpha \\ -{J_\lambda}^T \alpha \end{pmatrix}. \tag{81}$$

In [16] it is shown that η is indeed the normal vector of $F(P_{Q,\lambda})$ at $F(x, \lambda)$ for unconstrained PMOPs.

See Fig. 9 for an example of the normal at $(x, \lambda) = (0.0692, -0.0724, 0.5517)^T$ for PMOP1.

Having stated the normal vector η we are now in the position to compute the predictor direction $v_{(p)}$. For $v_{(p)}$ it should hold that the corresponding movement in objective space $Jv_{(p)}$ is along the linearized family of Pareto fronts at a given point $F(x_0, \lambda_0)$. That is, for a given (desired) direction $d \in \mathbb{R}^{k+l}$ one can compute the 'best fit' direction $d_{(p)}$ as follows: compute a QR-factorization of η,

$$\eta := QR = (q_1, q_2, \ldots, q_{k+l})R, \tag{82}$$

and define

$$Q_2 := (q_2, \ldots, q_{k+l}) \in \mathbb{R}^{(k+l)\times(k+l-1)}. \tag{83}$$

Then the orthogonal projection of d onto the orthogonal complement $\eta^\perp$ of η is given by

$$d_{(p)} := Q_2 Q_2^T d, \tag{84}$$

and $v_{(p)}$ can be computed via (27). Algorithm 1 puts together the above discussion.

The λ-DS continuation can now in principle be realized as the DS continuation for MOPs: first, a predictor is selected that performs a step along the linearized family of Pareto fronts. In a second step, the predicted point is then corrected back to the set of interest via an application of the λ-DS descent method.

In the following section, we go into detail for such a continuation method that performs a move in λ-space. Such methods are for instance interesting for certain online optimization problems [11, 40, 41].

A Continuation Method for a Movement in λ-Space. In the following we assume that we are not interested to move *along* the Pareto front as e.g. done for the DS

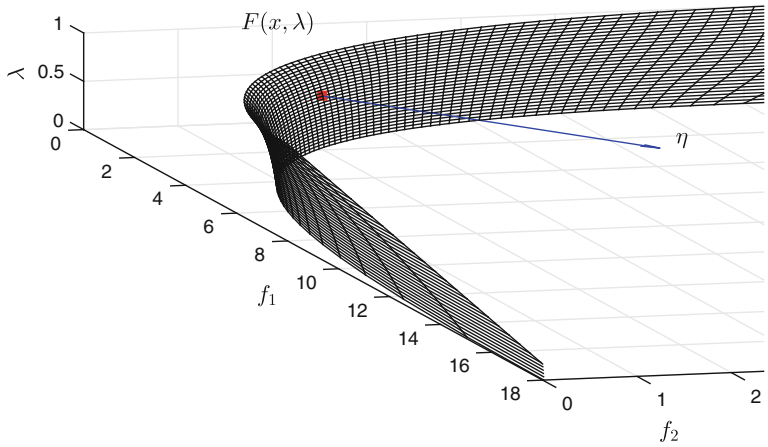


Fig. 9 Example of the normal vector η for PMOP1 for $(x, \lambda) = (0.0692, -0.0724, 0.5517)^T$

Algorithm 1 Algorithm Get Predictor Direction

Require: An initial KKT point (x_0, λ_0) with convex weight α, and search direction $d \in \mathbb{R}^k$.
Ensure: A direction $v_{(p)}$ s.t. $Jv_{(p)}$ points along the linearized set $F(P_{Q,\lambda})$ at $F(x_0, \lambda_0)$ and s.t. $Jv_{(p)}$ is the best fit solution of d.

1: $\eta := \begin{pmatrix} \alpha \\ -J_\lambda{}^T \alpha \end{pmatrix}$
2: $\eta := QR = (q_1, q_2, \ldots, q_{k+l})R$
3: $Q_2 := (q_2, \ldots, q_{k+l})$
4: $d_{(p)} := Q_2 Q_2^T d$
5: **return** $v_{(p)} := J^+ d_{(p)}$

continuation method, but to perform a movement in λ-space, i.e., orthogonal to the Pareto fronts. For the predictor direction $d_{pred} = (d_f, d_\alpha)$ we thus assume that

1. d_{pred} is perpendicular to the tangent space of Pareto front of F_λ, and
2. d_{pred} is perpendicular to the normal vector η.

Item 1 implies that d_f is orthogonal to $span\{\alpha\}^\perp$ by which follows that

$$d_f = \mu\alpha \quad \text{for a} \quad \mu \in \mathbb{R}. \tag{85}$$

By item 2 we obtain

$$\begin{aligned} 0 = \langle \eta, d \rangle &= \left\langle \begin{pmatrix} \alpha \\ -J_\lambda^T \alpha \end{pmatrix}, \begin{pmatrix} \mu\alpha \\ d_\lambda \end{pmatrix} \right\rangle \\ &= \mu \|\alpha\|^2 + \alpha^T J_\lambda^T d_\lambda. \end{aligned} \tag{86}$$

Thus, possible predictor directions are given by (without considering normalization nor orientation)

$$d_{pred} = \begin{pmatrix} \mu\alpha \\ d_\lambda \end{pmatrix}, \quad \mu \in \mathbb{R}, \tag{87}$$

where d_λ solves Eq. (86).

Remark 2 For the special case $l = 1$ (i.e., $\lambda \in \mathbb{R}$) there are two cases

1. If $\alpha^T J_\lambda \neq 0$, then d_λ is given by

$$d_\lambda = \frac{\mu ||\alpha||_2^2}{\alpha^T J_\lambda} \tag{88}$$

and thus we obtain (again without normalization nor orientation)

$$d_{pred} = \begin{pmatrix} \alpha \\ \|\alpha\|^2 / (\alpha^T J \lambda) \end{pmatrix}. \tag{89}$$

2. Else if $\alpha^T J_\lambda = 0$, then it follows by (86) that $\mu = 0$ and thus that also $d_f = 0$. Since d_{pred} is orthogonal to η it follows that

$$d_{pred} = \begin{pmatrix} 0 \\ d_\lambda \end{pmatrix}, \quad \text{where } d_\lambda \neq 0. \tag{90}$$

We can now formulate the continuation method that performs a movement in λ-space. Given a point (x_0, λ_0) with associated weight α, we compute in the first step the predictor as

$$p := (x_0, \lambda_0) + t v_{pred}, \tag{91}$$

where

$$v_{pred} = J^+ d_{pred} = \begin{pmatrix} v_{pred,x} \\ v_{pred,\lambda} \end{pmatrix} \tag{92}$$

is the search direction and $t > 0$ the chosen step size. For this, we adapt the step size control from the DS continuation method as follows: compute

$$t_i = \frac{\varepsilon}{\langle \nabla g_i, v_{pred} \rangle}, \quad i = 1, \ldots, k + l, \tag{93}$$

where $\varepsilon > 0$ represents a small (problem dependent) change in objective space. The value of t_i is chosen such that

$$\|g_i((x_0, \lambda_0) + t_i v_{ped}) - g_i(x_0, \lambda_0)\| \approx \varepsilon. \tag{94}$$

Finally, we choose t as the smallest of these step sizes,

$$t := \min_{i=1,\ldots,k+l} t_i, \tag{95}$$

in order not to obtain too large changes with respect to any of the objective functions.

Given a predictor point p, the next step is to correct this point to the solution curve. To accomplish this task, there are several possibilities. Two of them are

1. To use λ-DS descent method using $d_f = -\alpha$ and $d_\alpha = 0$, i.e., to use the corrector direction
$$d_{corr} = \begin{pmatrix} -\alpha \\ 0 \end{pmatrix}. \tag{96}$$

2. To solve the following minimization problem
$$\min_x \sum_{i=1}^{k} \alpha_i f_i(x, \lambda_1), \tag{97}$$
i.e., to solve the weighted sum problem [14] where the weights α are the KKT weights of (x_0, λ_0) and $\lambda_1 = \lambda_0 + t v_{pred,\lambda}$ is fixed to the λ-value of the predictor.

Algorithm 2 shows the pseudo code of the continuation method that is discussed above for PMOPs with $l = 1$.

Algorithm 2 Continuation method over λ-space using λ-DS for $\lambda \in \mathbb{R}$

Require: An initial solution (x_0, λ_0) with associated convex weight α, a threshold $\varepsilon \in \mathbb{R}_+$, and the number of iterations $ns \in \mathbb{N}_+$.
Ensure: A set of candidate solutions (x_i, λ_i), $i = 0, \ldots, ns$.
1: $i := 0$.
2: **while** $i < ns$ **do**
3: Compute d_λ as in Eq. (88).
4: Set d_{pred} as in Eq. (89).
5: Compute v as in Eq. (29).
6: Compute t using Eqs. (93) and (95).
7: $p_i = (\mathbf{x}_i, \lambda_i) - \text{sgn}(d_\lambda) t v$.
8: Compute $(\mathbf{x}_{i+1}, \lambda_{i+1})$ by using Eq. (97) or using Eq. (26) setting $d_f = -\alpha$ and $d_\lambda = 0$, using in both p_i as the initial point and fixing the value of α.
9: $i := i + 1$.
10: **end while**

Example 3 Figures 10, 11 and 12 and Tables 3, 4 and 5 show graphically and numerically the comparison and performance of the two λ-DS continuation methods proposed above. To perform both continuation methods it is mandatory that the starting point belongs to the solution set $(x_0, \lambda_0) \in P_{Q,\Lambda}$. The step size for the predictor point uses Eqs. (93) and (95) with $\varepsilon = 0.09$. In this case the stopping criterion is given by $\lambda < ub$ where ub is the upper bound of the domain of λ. As we use different correctors in each PC method, the solution curves also differ. The difference is most significantly for PMOP3.

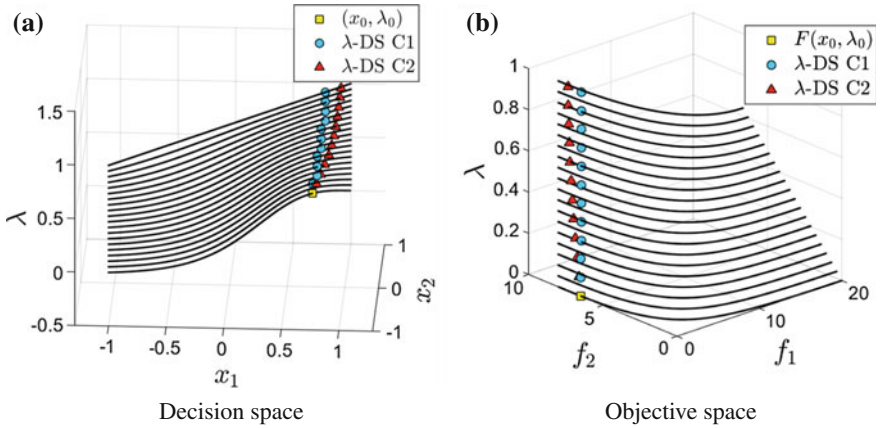


Fig. 10 Result of the two λ-DS continuation methods on PMOP1, using $(x_0, \lambda_0) = (0.671, 0.918, 0)$

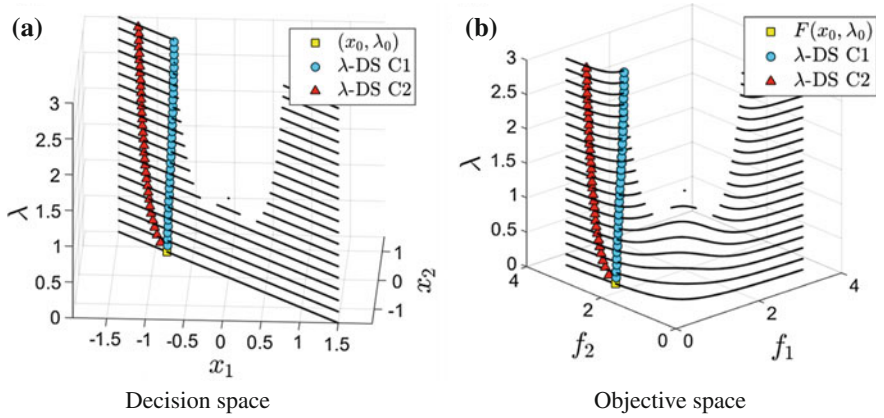


Fig. 11 Result of the two λ-DS continuation methods on PMOP2, using $(x_0, \lambda_0) = (-0.84, 0.84, 0)$

Example 4 The continuation method λ-DS C2 traces the implicitly defined curve[4] $H^{-1}(0)$ of the map

$$H : \mathbb{R}^{n+l} \to \mathbb{R}^n,$$
$$H(x, \lambda) = \sum_{i=1}^{k} \nabla_x f_i(x, \lambda). \tag{98}$$

In the following, we compare the performance of λ-DS C2 against 'classical' predictor–corrector (PC) methods applied on the zero set of H as e.g. described

[4]We consider in our computations only the case $l = 1$.

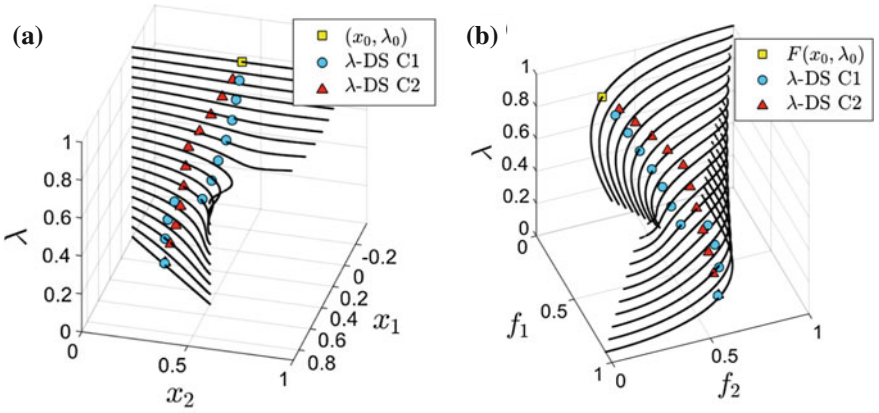


Fig. 12 Result of the two λ-DS continuation methods on PMOP3, using $(x_0, \lambda_0) = (0.558, 0.558, 0.5)$

Table 3 Number of objective function and Jacobian evaluations needed to perform λ-DS continuation method for PMOP 1

Information	λ-DS C1	λ-DS C2
# of $F(\mathbf{x}, \lambda)$	35	171
# of $J(\mathbf{x}, \lambda)$	14	11
# of iterations	11	11

Table 4 Number of objective function and Jacobian evaluations needed to perform λ-DS continuation method for PMOP 2

Information	λ-DS C1	λ-DS C2
# of $F(\mathbf{x}, \lambda)$	33	615
# of $J(\mathbf{x}, \lambda)$	33	33
# of iterations	33	33

in [1]. For such methods, the predictor direction is given by the last l column vectors of Q, where

$$H'(x, \lambda)^T = QR \tag{99}$$

is a QR-factorization of $H'(x, \lambda)^T$. In the corrector step, a solution of

$$H(x, \lambda) = 0 \tag{100}$$

that is 'close' to the predicted point p is computed e.g. via an application of a Gauss-Newton method starting with p. Thus, in this realization, in both steps the objectives' Hessians are required.

Table 5 Number of objective function and Jacobian evaluations needed to perform λ-DS continuation method for PMOP 3

Information	λ-DS C1	λ-DS C2
# of $F(\mathbf{x}, \lambda)$	333	590
# of $J(\mathbf{x}, \lambda)$	63	13
# of iterations	13	13

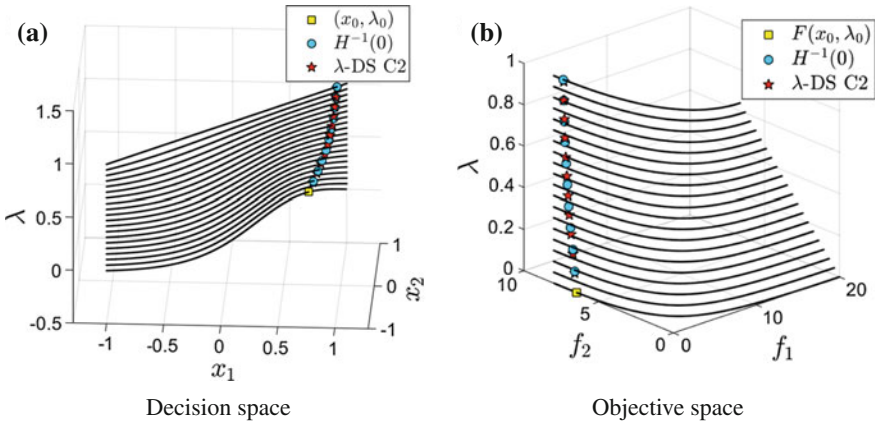


Fig. 13 Comparison of λ-DS C2 against the classical PC method on (98) for PMOP1, using $(x_0, \lambda_0) = (0.558, 0.558, 0.5)$

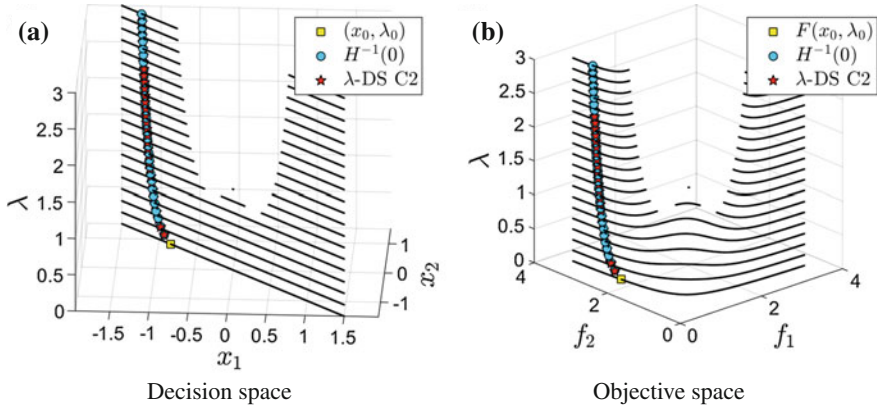


Fig. 14 Comparison of λ-DS C2 against the classical PC method on (98) for PMOP2, using $(x_0, \lambda_0) = (-0.84, 0.84, 0)$

Figures 13, 14 and 15 show the numerical results for the two continuation methods as well as the exact curves $H^{-1}(0)$ for PMOP1 to PMOP2 where we have used the same initial conditions as in the previous example. Here, in order to be fair, we have chosen the same step size as for the Example 3, using Eqs. (93) and (95)

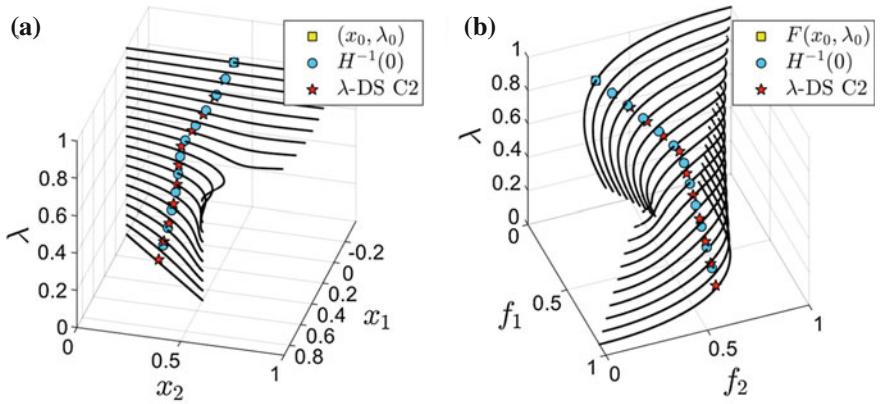


Fig. 15 Comparison of λ-DS C2 against the classical PC method on (98) for PMOP3, using $(x_0, \lambda_0) = (0.558, 0.558, 0.5)$

Table 6 Number of objective function, Jacobian, and Hessian matrix evaluations needed to perform λ-DS C2 and classical PC method for PMOP1

Information	PC	λ-DS C2
# of $F(\mathbf{x}, \lambda)$	98	171
# of $J(\mathbf{x}, \lambda)$	187	11
# of $H(\mathbf{x}, \lambda)$	98	0

Table 7 Number of objective function, Jacobian, and Hessian matrix evaluations needed to perform λ-DS C2 and classical PC method for PMOP2

Information	PC	λ-DS C2
# of $F(\mathbf{x}, \lambda)$	337	615
# of $J(\mathbf{x}, \lambda)$	635	33
# of $H(\mathbf{x}, \lambda)$	337	0

Table 8 Number of objective function, Jacobian, and Hessian matrix evaluations needed to perform λ-DS C2 and classical PC method for PMOP3

Information	PC	λ-DS C2
# of $F(\mathbf{x}, \lambda)$	251	99
# of $J(\mathbf{x}, \lambda)$	468	28
# of $H(\mathbf{x}, \lambda)$	251	0

with $\varepsilon = 0.09$. The figures show that both methods yield very similar and satisfying results compared to the exact solution curves while the cost of both methods differ significantly, compare to Tables 6, 7 and 8.

4 λ-DDS: A Gradient Free Realization of λ-DS

The key of λ-DS is to solve Eq. (26) in order to find a vector ν such that the search can be steered into the direction d in objective space. For this, the most expensive part might be the computation or the approximation of the objectives' gradients. In this section, we suggest an alternative way to compute such search directions ν without explicitly computing or approximating the Jacobian. Instead, the information of function values $F(y)$ of points y in the neighborhood of a given point x_0 are utilized for the approximation of ν. This method can be viewed as a particular forward difference (FD) method, however, this method has two crucial advantages over the classical Jacobian approximation via FD (e.g., [25]):

(i) fewer additional function evaluations are required to obtain a direction ν such that Eq. (7) is satisfied, and
(ii) existing neighborhood information can be utilized leading to a further reduction of the cost.

Item (ii) is in particular interesting in the context of set-based optimization strategies such as specialized evolutionary algorithms as ν can ideally be computed 'for free' (i.e., without any additional function evaluation).

The general idea of the λ-DDS ('discrete' λ-DS) is as follows: if a point $x_0 \in \mathbb{R}^n$ is given where the local search has to be performed according to a function $f : \mathbb{R}^n \to \mathbb{R}$ as well as another point $x_i \in \mathbb{R}^n$ together with its function value $f(x_i)$, then this information is already sufficient to approximate the directional derivative in direction

$$\nu_i := \frac{x_i - x_0}{\|x_i - x_0\|} \tag{101}$$

without spending additional function evaluations. More precisely, define the line search function

$$f_{\nu_i} : \mathbb{R} \to \mathbb{R}, \quad f_{\nu_i}(t) := f(x_0 + t\nu_i). \tag{102}$$

Then it holds for the directional derivative of f at x_0 in direction ν_i

$$f'_{\nu_i}(0) = \langle \nabla f(x_0), \nu_i \rangle = \frac{f(x_i) - f(x_0)}{\|x_i - x_0\|} + O(\|x_i - x_0\|), \tag{103}$$

where O denotes the Landau symbol. The approximation quality can be seen e.g. via considering the forward difference quotient on $f'_{\nu_i}(0)$.

Since by (27) $\nu_\lambda = d_\lambda$, we only have to approximate ν_f. For this, assume for now that we are given a candidate solution (x_0, λ_0) of (13) as well as the r directions $\nu_i \in \mathbb{R}^{n+l}$, $i = 1, \ldots, r$. Using

$$V := (\nu_1, \ldots, \nu_r) \in \mathbb{R}^{(n+l)\times r} \tag{104}$$

we obtain

$$\mathscr{F} := \begin{pmatrix} J_x & J_\lambda \end{pmatrix} V = \left(\langle \nabla g_i(x_0, \lambda_0), v_j \rangle \right)_{i=1,\dots,k,\ j=1,\dots,r} \in \mathbb{R}^{k \times r}. \tag{105}$$

Thus, every element m_{ij} of $\mathscr{F}$ is equal to the directional derivative of g_i at (x_0, λ_0) in direction v_j.

Given $\mathscr{F}$ and a direction d_f, a search direction v_f that satisfies (26) can now be computed first solving the linear system of equations

$$\begin{pmatrix} J_x & J_\lambda \end{pmatrix} V\mu = d_f \tag{106}$$

to obtain the vector $\mu \in \mathbb{R}^r$, and setting

$$v_x = V\mu. \tag{107}$$

If $r > k$, we suggest to proceed analog to (29) and to use

$$v_+^{(r)} = V(\begin{pmatrix} J_x & J_\lambda \end{pmatrix} V)^+ d_f. \tag{108}$$

So far, the method is still utilizing gradient information. To overcome this, we can use the above discussion: given a set (x_i, λ_i), $i = 1, \dots, r$, of neighboring solutions of (x_0, λ_0), we can set the search directions as

$$v_i := \frac{(x_i, \lambda_i) - (x_0, \lambda_0)}{\|(x_i, \lambda_i) - (x_0, \lambda_0)\|}, \quad i = 1, \dots, r, \tag{109}$$

and approximate the entries m_{ij} of $\mathscr{F}$ by

$$m_{ij} = \langle \nabla g_i(x_0, \lambda_0), v_j \rangle \approx \frac{g_i(x_j, \lambda_j) - g_i(x_0, \lambda_0)}{\|(x_j, \lambda_j) - (x_0, \lambda_0)\|}, \quad i = 1, \dots, k,\ j = 1, \dots, r. \tag{110}$$

Example 5 Consider the setting of Example 1. As the given problem is a MOP, the search direction is given by [34]

$$v_+^{(r)} = V(JV)^+ d, \tag{111}$$

where J is as in (51). When choosing the direction vectors v_i as the 10-dimensional canonical vectors e_i for $i = 1, \dots, r$ we obtain $v_+^{(2)} = \bar{v}_1$ (as defined in (54)), $v_+^{(4)} = \bar{v}_2$, …, $v_+^{(10)} = \bar{v}_5 = v_+$. That is, when choosing

$$x_i = x_0 + \bar{t} \frac{v_i}{|v_i\|} \tag{112}$$

as in (57) we obtain the same behavior as displayed in Fig. 5. In particular, we obtain better performances for increasing values of r.

In general, it can be expected that larger values of r lead to better results in terms of the movement in direction d, however, coming with the potential cost of further samplings. This is why it seems to be wise to include the given neighborhood information (e.g., if λ-DDS is applied within a MOEA) as this comes without additional cost. As a rule of thumb we have observed that $r \approx 0.4n$ leads to good results (the value of l is typically low and can be neglected here). We stress that nearly identical directions have to be avoided as else the condition number of V will get large leading to instabilities in the numerical solution of (111). However, even if neighboring information exists, this may not be sufficient and further samples have to be taken into account. In the following we discuss how such further samples can be computed.

Assume we are given $(x_0, \lambda_0) \in \mathbb{R}^{n+l}$ as well as the m neighboring[5] solutions (x_i, λ_i), $i = 1, \ldots, m$, and that the remaining $r - m$ solutions have to be sampled. Analog to the DS method for MOPs [34] it seems to be wise to compute these samples such that the difference vectors $(x_i, \lambda_i) - (x_0, \lambda_0)$, $i = m + 1, \ldots, r$, are orthogonal to (i) each other as well as (ii) to the previous difference vectors. For this, one can proceed as follows: compute a QR-factorization of $V := (v_1, \ldots, v_l)$, i.e.,

$$V = QR = (q_1, \ldots, q_l, q_{m+1}, \ldots, q_{n+l}), \tag{113}$$

where $Q \in \mathbb{R}^{(n+l)\times(n+l)}$ is an orthogonal matrix and $R \in \mathbb{R}^{(n+m)\times l}$ is a (generalized) right upper triangular matrix. Then it follows that

$$v_i \in span\{q_1, \ldots, q_i\}, \quad i = 1, \ldots, m, \tag{114}$$

and thus that

$$\langle v_i, q_j \rangle = 0, \quad \forall i \in \{1, \ldots, l\},\ j \in \{m + 1, \ldots, r\}. \tag{115}$$

One can thus compute the new samples via

$$\begin{aligned} v_{m+i} &= q_{m+i}, \quad i = 1, \ldots, r - m, \\ (x_{m+1}, \lambda_{l+1}) &= (x_0, \lambda_0) + t_{m+i} v_{m+i}, \quad i = 1, \ldots, r - m, \end{aligned} \tag{116}$$

where t_{m+i} is a given (problem dependent) step size.

The cost for the QR-factorization is $O((n + l)^3)$ in terms of time complexity, thus, relatively high. If $n + l$ is large and $r - m$ is small one can e.g. use the Gram-Schmidt procedure [25] to obtain the remaining sample points leading to a cost of $O((m - r)^2(n + l))$. For instance, if v_1 and $\bar{v}_2$ are given such that the vectors $\{v_1, \bar{v}_2\}$ are linearly independent, the second search direction can be computed by

$$v_2 := \bar{v}_2 - \langle v_1, \bar{v}_2 \rangle v_1. \tag{117}$$

[5] Diverse neighborhood relationships can be established, in this work we induced it through the Euclidean distance.

Algorithm 3 puts together the above discussion for the computation of a search vector v such that $Jv \approx d$ out of given neighborhood information.

Algorithm 3 Algorithm λ-DDS

Require: initial point (x_0, λ_0), neighborhood $N(x_0, \lambda_0)$ of (x_0, λ_0), search direction $d = (d_f, d_\lambda)^T$, number r of test points with $r \geq k$.
Ensure: direction $v \in \mathbb{R}^{10}$ with $Jv \approx d$
choose $(x_1, \lambda_1), \ldots, (x_r, \lambda_r) \in N(x_0, \lambda_0)$ such that the difference vectors $(x_i - x_0, \lambda_i - \lambda_0)$, $i = 1, \ldots, r$ are linear independent
for $i = 1, \ldots, r$ **do**
  $v_i := \frac{(x_i, \lambda_i) - (x_0, \lambda_0)}{\|(x_i, \lambda_i) - (x_0, \lambda_0)\|}$
end for
$V := (v_1, \ldots, v_r) \in \mathbb{R}^{(n+l) \times r}$
for $i = 1, \ldots, k$ **do**
  for $j = 1, \ldots, r$ **do**
    $m_{ij} = \frac{g_i(x_j, \lambda_j) - g_i(x_0, \lambda_0)}{\|(x_j, \lambda_j) - (x_0, \lambda_0)\|}$
  end for
end for
$M := \left(m_{ij}\right)_{i=1,\ldots,k,\ j=1,\ldots,r} \in \mathbb{R}^{k \times r}$
solve $M\mu = d_f$ to obtain μ
$v_x := V\mu$
$v_\lambda = d_\lambda$
return $v := (v_x, v_\lambda)$

Example 6 Consider PMOP4 in Table 10 using λ-DDS. In the following, we show the performance of the λ-DDS with different values of r. Figure 16 shows the produced movement toward the Pareto front. For $r = 4$ the method approaches the Pareto front

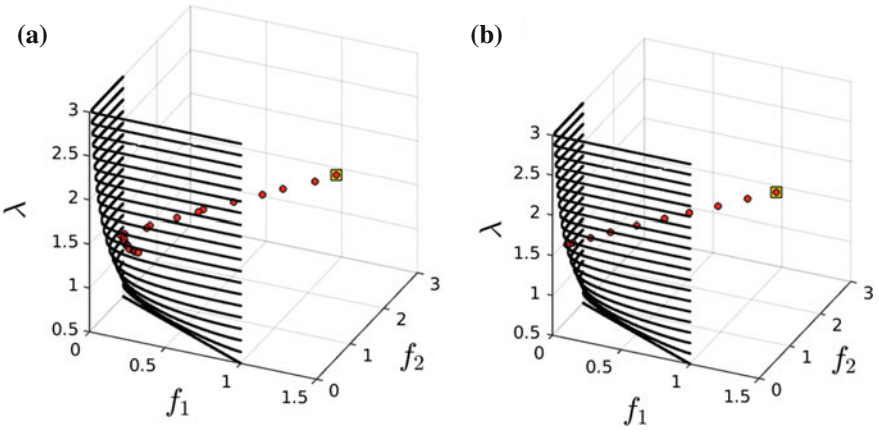


Fig. 16 λ-DDS on PMOP4 with $r = 4$ and $r = 5$ using $(x_0, \lambda_0) = (1, 1, 1, 1, 1, -1, -1, -1, -1, -1, 1.5)^T$

directly, however, in the case of $r = 5$ the trajectory looks more accurate. The number of function evaluations used in each subfigure were 456 and 550, respectively. This example takes into account the number of generated neighbors, nevertheless, in the case of evolutionary algorithms these neighbors typically could come for free.

5 On the Behavior of Stochastic Local Search Within PMOPs

The approach of the λ-DS can be used to a certain extent to understand the behavior of stochastic local search (SLS) within PMOPs which we will investigate in this section.

We will in the following demonstrate the relation of SLS with λ-DS, will further on discuss SLS within PMOPS at different stages of the (stochastic) search, and will finally make an attempt to explain the effect of SLS within set based stochastic search.

5.1 Relation of SLS and λ-DS

SLS refers here to the random selection of an offspring z_1 from the neighborhood of a given candidate z_0. To see the relation between SLS and λ-DS, assume that a point $z_1 = (x_1, \lambda_1)$ is chosen at random from a small neighborhood N of the point $z_0 = (x_0, \lambda_0)$. Note that then z_1 can be written as

$$z_1 = z_0 + 1(z_1 - z_0) = z_0 + ||z_1 - z_0|| \frac{z_1 - z_0}{||z_1 - z_0||}. \tag{118}$$

That is, the selection of z_1 can be viewed as a search in direction

$$v := \frac{z_1 - z_0}{||z_1 - z_0||} \tag{119}$$

and step size

$$t = ||z_1 - z_0||. \tag{120}$$

From Eq. (75) it can be seen that the movement in objective space when moving from z_0 in direction $z_1 - z_0$ in decision space for infinitesimal steps is given by $J(z_0)v$, i.e., we have

$$\frac{F(z_1) - F(z_0)}{||z_1 - z_0||} \approx J(z_0)v \tag{121}$$

if N is chosen to be small enough. Thus, the approach of λ-DS can be used to explain the respective movements in objective space when applying SLS to given points z_0.

5.2 SLS at Different Stages of the Search

Based on these considerations, we now consider three different scenarios for SLS within PMOPs that occur in different stages within a stochastic search algorithm such as an evolutionary strategy.

(a) $(\mathbf{x}, \lambda)$ 'far away' from $P_{Q,\Lambda}$.

Here we use an observation made in [6] for classical MOPs namely that the objectives' gradients may point into similar directions when the decision point (x, λ) is far from the Pareto set. We assume here for simplicity the extreme case namely that all gradients point into the same direction. For this, let

$$g := \nabla_x f_1(x, \lambda) \tag{122}$$

and assume that

$$\nabla_x f_i(x, \lambda) = \mu_i g, \quad i = 1, \ldots, k, \tag{123}$$

where $\mu_i > 0$ for $i = 1, \ldots, k$. Then

$$J_x v_x = \begin{pmatrix} \nabla_x f_1(x, \lambda)^T v_x \\ \vdots \\ \nabla_x f_k(x, \lambda)^T v_x \end{pmatrix} = g^T v_x \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_k \end{pmatrix}. \tag{124}$$

That is, the movement is 1-dimensional regardless of v_x which is n-dimensional. Since $J_x v_x = 0$ if and only if $v_x \perp g$, the probability is one that for a randomly chosen v_x either dominated or dominating solutions are found (and in case a dominated solution is found, the search has simply to be flipped to find dominating solutions).
Thus, for $v_\lambda = 0$, which means that the value of λ is not changed in the local search, we obtain for $\mu = (\mu_1, \ldots, \mu_k)^T$ the direction

$$Jv = \begin{pmatrix} g^T v_x \mu \\ 0 \end{pmatrix}. \tag{125}$$

Remark 3 For $v_\lambda \neq 0$, i.e., in the case that the value of λ is changed within the local search, no such physical meaning exists to the best of our knowledge. Nevertheless, the investigation of this problem will be one topic for future research.

(b) (x, λ) 'near' to $P_{Q,\Lambda}$.

Here we consider again the extreme case, namely that $\mathbf{x}$ is a KKT point of F_λ. That is, there exists a convex weight $\alpha \in \mathbb{R}^k$ such that

$$\sum_{i=1}^{k} \alpha_i \nabla_x f_i(\mathbf{x}, \lambda) = J_x^T \alpha = 0. \tag{126}$$

As the normal vector η is as in (81) we obtain

$$\langle J\nu, \eta \rangle = \langle \nu, J^T \eta \rangle = \left\langle \nu, \begin{pmatrix} J_x^T & 0 \\ J_\lambda^T & I_l \end{pmatrix} \begin{pmatrix} \alpha \\ -J_\lambda^T \alpha \end{pmatrix} \right\rangle = \left\langle \nu, \begin{pmatrix} J_x^T \alpha \\ J_\lambda^T \alpha - J_\lambda^T \alpha \end{pmatrix} \right\rangle = 0. \tag{127}$$

That is, it is either (i) $J\nu = 0$ or (ii) $J\nu$ is a movement orthogonal to η and thus along the linearized set at $F(\mathbf{x}, \lambda)$. If we assume that the rank of J_x is $k-1$, then the rank of J is $k-1+l$ and the dimension of the kernel of J is $n-k+l$. Hence, for a randomly chosen ν the probability is 1 that event (ii) happens.

Equation (127) tells us that the movement is orthogonal to the normal vector, but it remains to investigate in which direction of the tangent space the movement is performed. For this, let

$$\eta = QR = (q_1, q_2, \ldots, q_{k+l})R \tag{128}$$

be a QR factorization of η. Then, the vectors $q_2, \ldots, q_{k+l}$ form an orthonormal basis of the tangent space. If we assume again that the rank of J_x is $k-1$, then the rank of J is $k-1+l$. Since by Eq. (127) η is not in the image of J, there exist vectors $\nu_q, \ldots, \nu_{k+l}$ such that

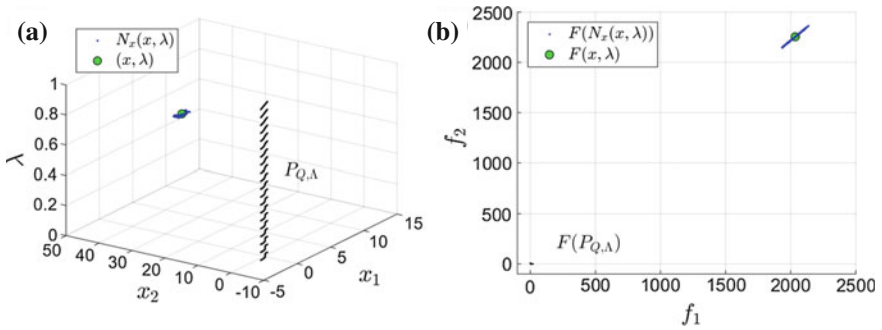
$$J\nu_i = q_i, \quad i = 2, \ldots, k+l. \tag{129}$$

Thus, a movement via SLS can be performed in all directions of the linearized family of Pareto fronts (i.e., both in $\mathbf{x}$- and λ-direction).

(c) (x, λ) 'in between'.

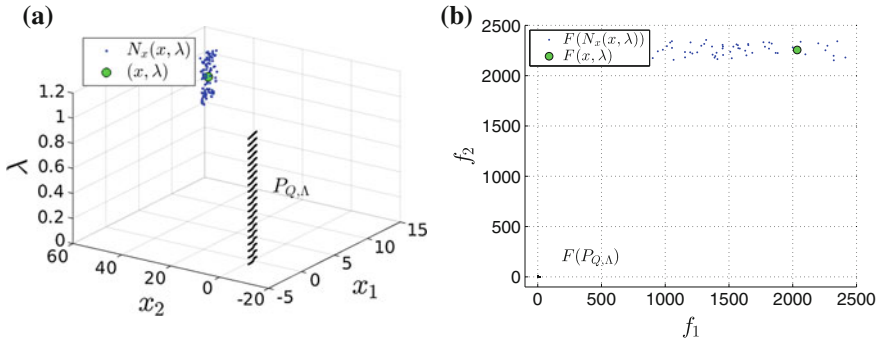
Apparently, points $(\mathbf{x}, \lambda)$ do not have to be far away from nor near to the set of interest but can be 'in between'. In this case, no clear preference of the movement in objective space can be detected. However, this 'opening' of the search compared to the 1-dimensional movement in early stages of the search is a very important aspect since it allows in principle to find (in the set based context and given a suitable selection mechanism) and spread the solutions. For instance, it may allow for finding multiple connected components. Note that under this light it even seems to be advantageous if some initial solutions are 'far' from the solution set.

Example 7 (a) We consider the sampling of neighboring solutions of candidate points (x, λ) in the three different stages (i.e., distances to the solution set) for PMOP1. Figures 17 and 18 show the behavior of SLS for 100 uniformly randomly chosen points near $(x, \lambda) = (10, 45.2, 0.7)^T$ for $\nu_\lambda \neq 0$ and $\nu_\lambda = 0$.



Decision space Objective space

Fig. 17 SLS on PMOP1 for a point that is 'far away' from $P_{Q,\Lambda}$ using $\nu_\lambda = 0$



Decision space Objective space

Fig. 18 SLS on PMOP1 for a point that is 'far away' from $P_{Q,\Lambda}$ using $\nu_\lambda \neq 0$

As neighborhood we have chosen

$$N_{(r_x,r_\lambda)}(x,\lambda) := \{(\hat{x},\hat{\lambda}) \in \mathbb{R}^{n+l} \ : \ \|x-\hat{x}\| \leq r_x \quad \text{and} \quad \|\lambda-\hat{\lambda}\| \leq r_\lambda\} \quad (130)$$

with radius $r_x = 2$ in x-space and $r_\lambda = 0.3$ (respectively $r_\lambda = 0$) in λ-space. For the case $\nu_\lambda = 0$ a clear movement toward/against $F(P_{Q,\Lambda})$ can be observed while this is not the case for $\nu_\lambda \neq 0$. Thus, it may make sense to exclude the change of the value of λ in early stages of the search process where the individuals of the populations are supposed to be far away from the set of interest.

Figure 19 shows an example for $(x,\lambda) = (0.44, 0.47, 0.84)^T$ which is near the solution set, and $r_x = r_\lambda = 0.2$. Again, by construction, no structure in decision space can be observed, but a clear movement along $F(P_{Q,\Lambda})$ can be seen in objective space.

Finally, Fig. 20 shows the neighborhood sampling for $(\mathbf{x},\lambda) = (1,-1,0.5)^T$ which is 'in between' using $r_x = r_\lambda = 0.2$. A movement in all directions can be seen, now both in decision and objective space.

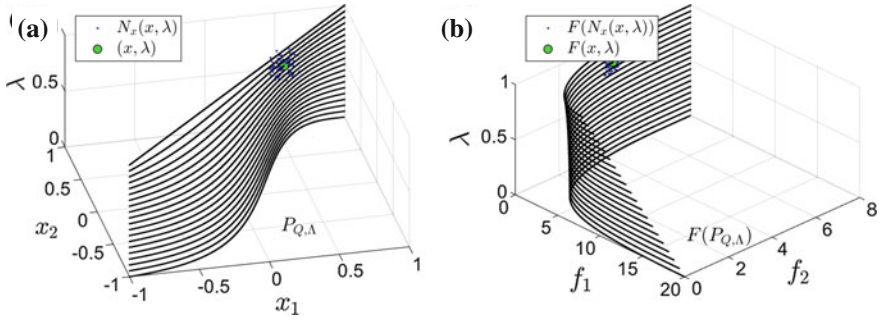


Fig. 19 SLS on PMOP1 for a point that is 'near' to $P_{Q\Lambda}$

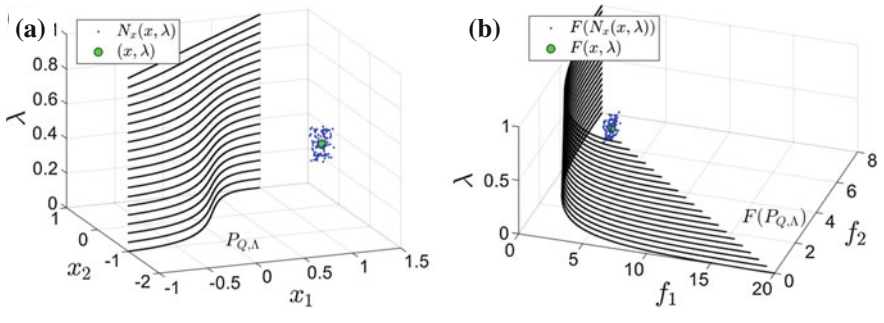


Fig. 20 SLS on PMOP1 for a point that is 'in between' using $\nu_\lambda \neq 0$

(b) Now, we show analog results of the SLS for PMOP3. Figure 21 depicts the behavior of SLS for 100 uniformly randomly chosen points in the neighborhood of $(x, \lambda) = (10.0, 10.0, 0.7)^T$ with $r_x = 1$ and $r_\lambda = 0$. We observed the same movement as PMOP1, which means that the produced points are moving toward and against $F(P_{Q,\Lambda})$.
Next, in Fig. 22, the point $(x, \lambda) = (0.1098, 0.4146, 0.5789)$ which is 'near' to the solution set is chosen for SLS with $r_x = 0.11$ and $r_\lambda = 0.1$. No movement is observed in decision space, however, a movement along $F(P_{Q,\Lambda})$ can be detected. In the case of Fig. 23, we sample around the same 100 points over the neighborhood of $(x, \lambda) = (0.4, -0.5, 0.5)^T$, such a point is 'in between' not near nor far away from the solution set. The size of the neighborhood was given by $r_x = 0.11$ and $r_\lambda = 0.1$. Here again, we observed a movement in all directions in decision as in objective space.

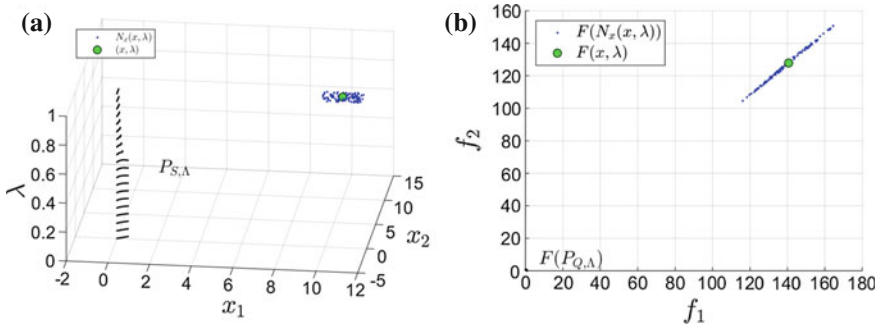


Fig. 21 SLS on PMOP3 for a point that is 'far away' from $P_{Q,\Lambda}$ using $\nu_\lambda = 0$

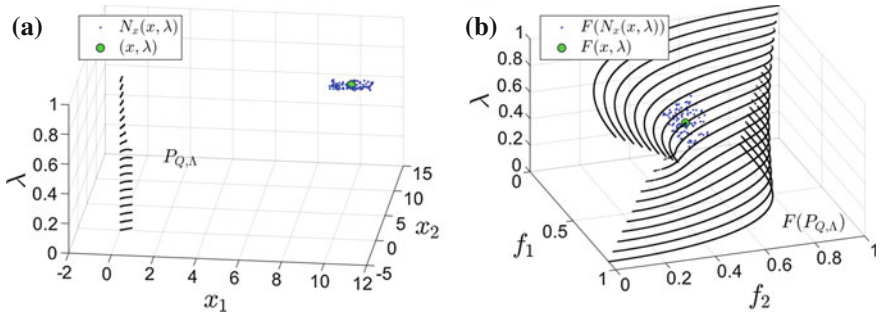


Fig. 22 SLS on PMOP2 for a point that is 'near' to $P_{Q,\Lambda}$

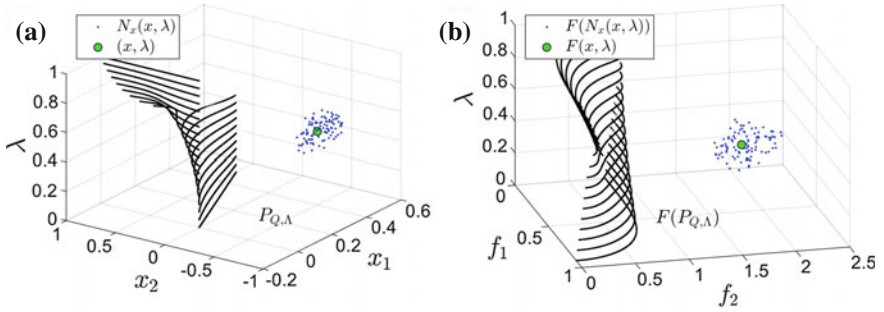


Fig. 23 SLS on PMOP3 for a point that is 'in between' using $\nu_\lambda \neq 0$

5.3 SLS Within Set Based Search

As next step we investigate the influence of SLS within set based methods. In order to prevent interferences with other effects we have thus to omit all other operators, e.g.,

crossover and all other 'swarm-like' strategies. The Simple Neighborhood Search (SNS) for PMOPs takes this into consideration: initially, a generation $A_0 \subset \mathbb{R}^{n+l}$ is chosen at random, where Λ is discretized into

$$\tilde{\Lambda} = \{\lambda_1, \ldots, \lambda_s\}. \tag{131}$$

In the iteration process, for every element $(a_x, a_\lambda) \in A_i$, a new element (b_x, b_λ) is chosen via SLS, where b_λ has to take one of the values of $\tilde{\Lambda}$ (which we properly adjusted). The given archive A_i and the set of newly created solutions B_i are the basis for the sequences of candidate solutions $A_i^l, l = 1, \ldots, s$, and the new archive A_{i+1}: for A_i^l a selection of $A_i \cup B_i$ with λ-value λ_l is taken, where we use *ArchiveUpdateTight*2 (Algorithm 5, see also [33]) to update the archives. A_{i+1} is the union of these sets (plus the respective λ values). Algorithm 4 shows the pseudo code of SNS. Given an archive $A \subset \mathbb{R}^{n+l}$, then denote by

$$\pi(A, \lambda_i) := \{a \in \mathbb{R}^n \ : \ (a, \lambda_i) \in A\} \tag{132}$$

the set of x-values of A with λ-value λ_i. It is, for instance,

$$\pi(P_{Q,A}, \lambda_i) = P_{Q,\lambda_i}, \tag{133}$$

i.e., the Pareto set of F_{λ_i}. Finally, given an archive $B \subset \mathbb{R}^n$, then

$$(B, \lambda) := \{(b, \lambda) \ : \ b \in B\} \subset \mathbb{R}^{n+l} \tag{134}$$

defines an archive in the compound (x, λ)-space.

Algorithm 4 SNS for PMOPs

Require: Neighborhood $N_i(x, \lambda)$ of a given point (x, λ) in iteration i, and number of iterations *itermax*.
Ensure: Sequence A_i^l of candidate solutions for $F_{\lambda_l}, l = 1, \ldots, s$

Generate $A_0 \subset \mathbb{R}^{n+l}$ at random
for $i = 0, 1, 2, \ldots,$ *itermax* **do**
  $B_i^l := \emptyset$
  for all $(a_x, a_\lambda) \in A_i$ **do**
    choose $(b_x, b_\lambda) \in N_i(a_x, a_\lambda)$
    $B_i := B_i \cup (b_x, b_\lambda)$
  end for
  $A_i^l := \pi(A_i, \lambda_l), l = 1, \ldots, s$
  $B_i^l := \pi(B_i, \lambda_l), l = 1, \ldots, s$
  $A_{i+1}^l := \mathit{ArchiveUpdateTight}2(B_i^l, A_i^l), l = 1, \ldots, s$
  $A_{i+1} := \bigcup_{l=1}^s (A_{i+1}^l, \lambda_l)$
end for

Algorithm 5 $A := ArchiveUpdateTight2(P, A_0)$

Require: initial archive A_0, candidate set P, threshold ε in objective space, threshold $\tilde{\Delta}$ in decision space, safety factor $\Theta_\varepsilon \in (0, 1)$.
Ensure: updated archive A

```
A := A_0
for all p ∈ P do
    if ∄a ∈ A : a ≺_{Θε} p, or(∄a ∈ A : a ≺ p and ∀a ∈ A : d_∞(F(a), F(p)) > Δ̃) then
        A := A ∪ {p}
    end if
    for all a ∈ A do
        if p ≺ a then
            A := A ∪ {p}\{a}
        end if
    end for
end for
return A
```

In order to test SNS we apply it to five different PMOPs, the problems PMOP1 to PMOP3 as defined above as well as to PMOP4 and PMOP5 shown in Table 10. For sake of a comparison we consider a simple global search (GS) mechanism that uniformly at random picks elements from the domain which are updated by *ArchiveUpdateTight*2. The domains for all models can be seen in Table 9.

It remains to measure the performance of the two algorithms with respect to their approximation of $P_{Q,\Lambda}$ and $F(P_{Q,\Lambda})$, respectively. Until now, no such indicator exists to the best of our knowledge. Since we are discretizing in this study Λ via $\tilde{\Lambda}$, we have decided to take the power mean of the Δ_p values for the candidate sets and the Pareto sets/fronts for the different values of $\tilde{\Lambda}$ ('slices'). More precisely, we define the averaged Hausdorff distance of the solution set $P_{Q,\lambda}$ and the archive A as

$$\Delta_p(P_{Q,\lambda}, A) := \left(\frac{1}{s} \sum_{i=1}^{s} \Delta_p(\pi(A, \lambda_l), P_{Q,\lambda_i})^p \right)^{1/p}, \tag{135}$$

where $\Delta_p(A, B)$ denotes the averaged Hausdorff distance between the sets A and B (see also Definition 2). Analogously, the averaged distance between $F(P_{Q,\lambda})$ and $F(A)$ is defined by

Table 9 Domains for the test problems. λ takes 20 uniformly distributed values ('slices')

Problem	Domain
PMOP1	$x \in [0, 1]^2$, $\lambda \in [0, 1]$
PMOP2	$x \in [-1.5, 1.5]^2$, $\lambda \in [0, 3]$
PMOP3	$x \in [-3, 3]^2$, $\lambda \in [0, 1]$
PMOP4	$x \in [-10, 10]^2$, $\lambda \in [0.5, 3]$
PMOP5	$x \in [0, 1]^2$, $\lambda \in [120, 130]$

Table 10 Test problems

PMOP1 (Convex fronts—2D)
$F_\lambda : \mathbb{R}^2 \to \mathbb{R}^2$
$F_\lambda(x) := (1-\lambda)F_1(\mathbf{x}) + \lambda F_2(\mathbf{x})$
$F_1(x_1, x_2) = \begin{pmatrix} (x_1-1)^4 + (x_2-1)^2 \\ (x_1+1)^2 + (x_2+1)^2 \end{pmatrix}$
$F_2(x_1, x_2) = \begin{pmatrix} (x_1-1)^2 + (x_2-1)^2 \\ (x_1+1)^2 + (x_2+1)^2 \end{pmatrix}$
PMOP2 (Convex-concave to disconnected fronts—2D)
$f_1, f_2 : \mathbb{R}^3 \to \mathbb{R}$
$f_1(x, \lambda) = \frac{1}{2}(\sqrt{1+(x_1+x_2)^2} + \sqrt{1+(x_1-x_2)^2} + x_1 - x_2) + \lambda \cdot e^{-(x_1-x_2)^2}$
$f_2(x, \lambda) = \frac{1}{2}(\sqrt{1+(x_1+x_2)^2} + \sqrt{1+(x_1-x_2)^2} - x_1 + x_2) + \lambda \cdot e^{-(x_1-x_2)^2}$
PMOP3 (Convex to concave fronts—2D)
$F_\lambda : \mathbb{R}^2 \to \mathbb{R}^2$
$F_\lambda(x) := (1-\lambda)F_1(x) + \lambda F_2(x)$
$F_1(\mathbf{x}) = \begin{pmatrix} (x_1^2 + x_2^2)^{0.125} \\ ((x_1-0.5)^2 + (x_2-0.5)^2)^{0.25} \end{pmatrix}$
$F_2(\mathbf{x}) = \begin{pmatrix} x_1^2 + x_2^2 \\ (x_1-a_1)^2 + (x_2-a_2)^2 \end{pmatrix}$
PMOP4 (Convex to linear fronts—2D)
$f_1, f_2 : \mathbb{R}^3 \to \mathbb{R}$
$f_1(x, \lambda) = \frac{(\sum_{i=1}^n x_i^2)^\lambda}{n^\lambda}$
$f_2(x, \lambda) = \frac{(\sum_{i=1}^n (1-x_i)^2)^\lambda}{n^\lambda}$
PMOP5 (Convex fronts, non-linear Pareto set—2D)
$f_1, f_2 : \mathbb{R}^3 \to \mathbb{R}$
$f_1(x, \lambda) = x_1$
$f_2(x, \lambda) = g * \left(1 - \sqrt{\frac{f_1}{g}}\right)$
$g(x, \lambda) = 1 + \sum_{i=2}^n \left(x_i - \left(\sin(0.5 * \pi * \frac{\lambda}{nT})\right)\right)^2$

$$\Delta_p(F(P_{Q,\lambda}), F(A)) := \left(\frac{1}{s} \sum_{i=1}^{s} \Delta_p(F_{\lambda_i}(\pi(A, \lambda_l)), F(P_{Q,\lambda_i}))^p \right)^{1/p}. \tag{136}$$

The indicators are thus straightforward adaptions of Δ_p to the context of PMOPs. However, a more in-depth discussion is certainly desired which we have to leave for future work.

Table 11 shows the resulting Δ_p values for the final archives A_f for SNS and GS on the test problems and for a budget of 5,000 function evaluations. As it can be seen, SNS significantly outperforms GS in terms of Δ_p both in decision and objective space in 4 out of the 5 problems. An exception is PMOP2, where GS outperforms SNS (also significantly). Reasons for this seems to be that the decision variable bounds for PMOP2 are relatively small, and 5,000 randomly chosen candidates within this set seem to be enough to obtain a good approximation of the solution set. Figures 25,

Table 11 Comparison of SNS and GS on the test functions and a budget of 5,000 function evaluations. The Δ_2 values are averaged over 20 independent runs. Noticed that Δ_2 is related to the Euclidean distance

	$\Delta_p(P_{Q,\lambda}, A_f)$				$\Delta_p(F(P_{Q,\lambda}), F(A_f))$			
	SNS		GS		SNS		GS	
Problem	Mean	Std	Mean	Std	Mean	Std	Mean	Std
PMOP1	**0.161**	0.039	0.813	0.055	**1.385**	0.325	7.059	2.888
PMOP2	0.683	0.250	**0.231**	0.030	0.687	0.255	**0.245**	0.042
PMOP3	**0.186**	0.030	0.543	0.042	**0.206**	0.025	0.568	0.092
PMOP4	**0.106**	0.050	0.740	0.076	**0.193**	0.040	3.311	2.352
PMOP5	**0.140**	0.066	0.407	0.017	**0.045**	0.003	0.415	0.030

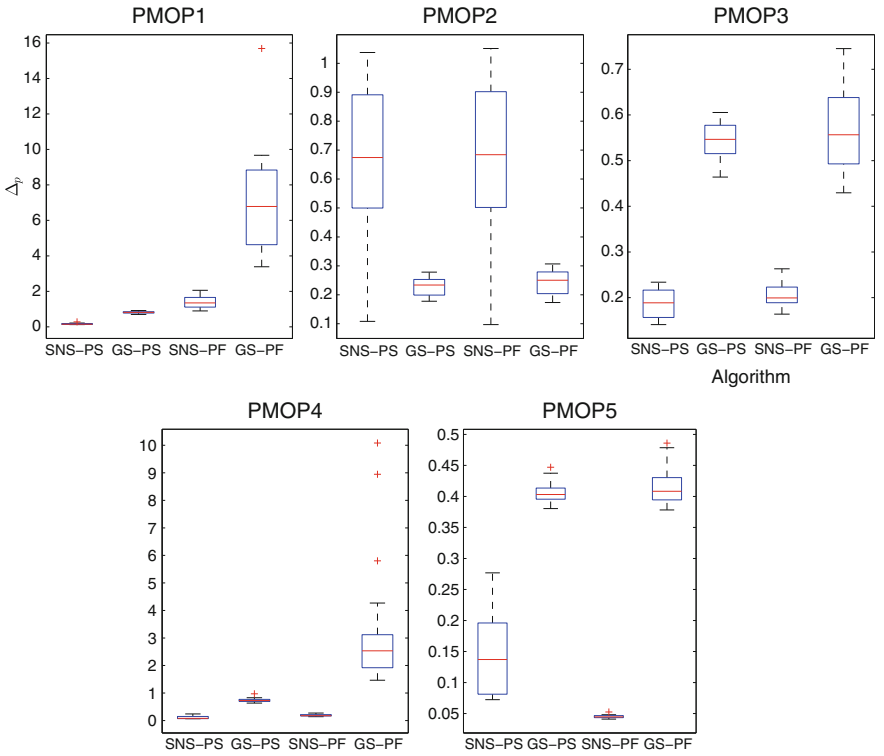


Fig. 24 Boxplots comparison of the SNS and GS algorithm over the five test functions in both decision (PS) and objective space (PF)

26, 27, 28 and 29 show the final archives from both methods on PMOP1 to PMOP5. Figure 24 shows the representative boxplots of the experiments.

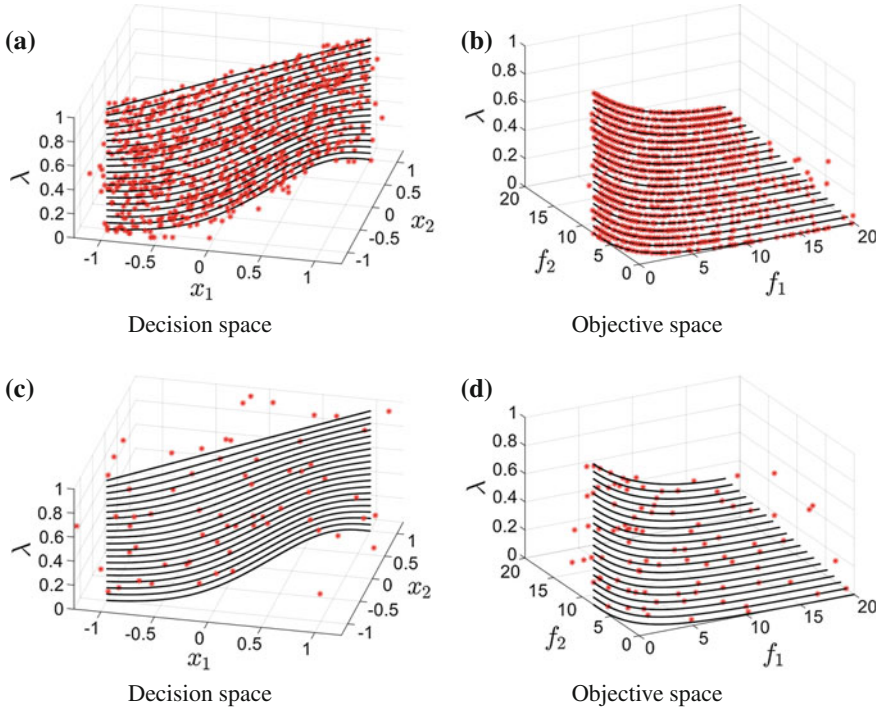


Fig. 25 Results for SNS (*above*) and GS (*below*) for PMOP1

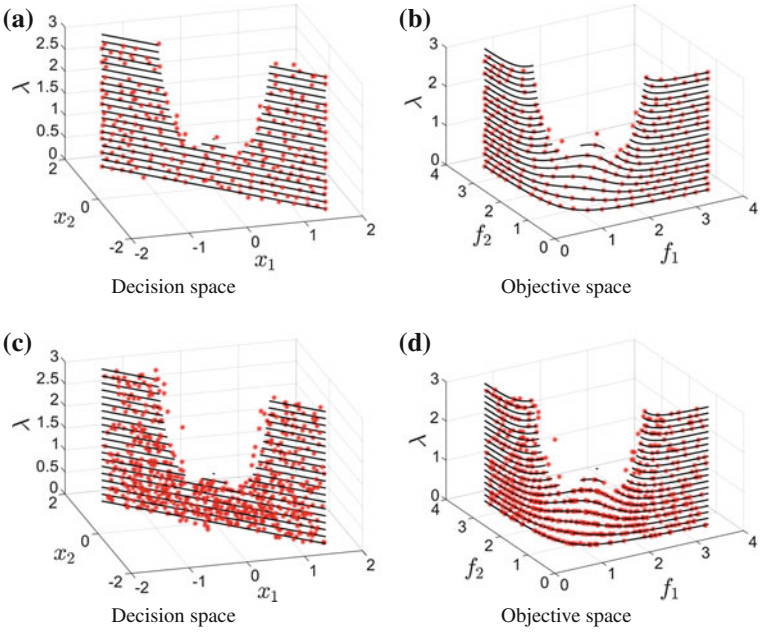


Fig. 26 Results for SNS (*above*) and GS (*below*) for PMOP2

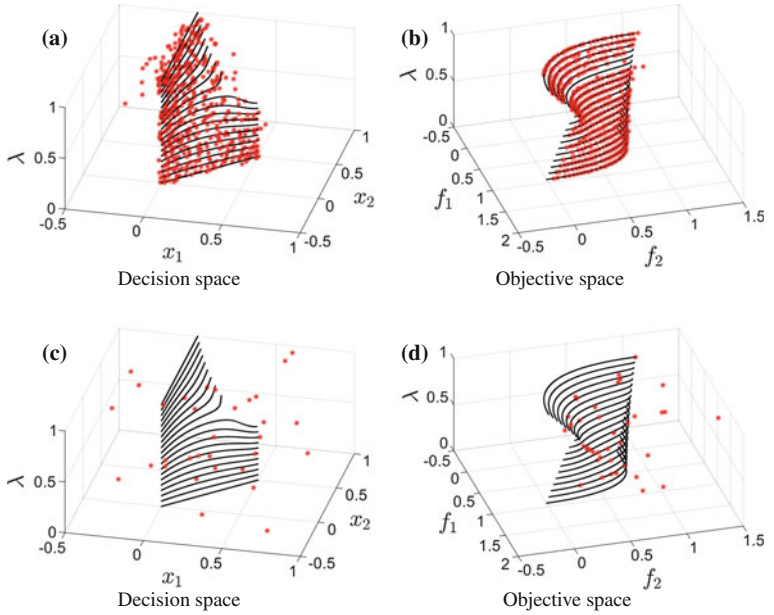


Fig. 27 Results for SNS (*above*) and GS (*below*) for PMOP3

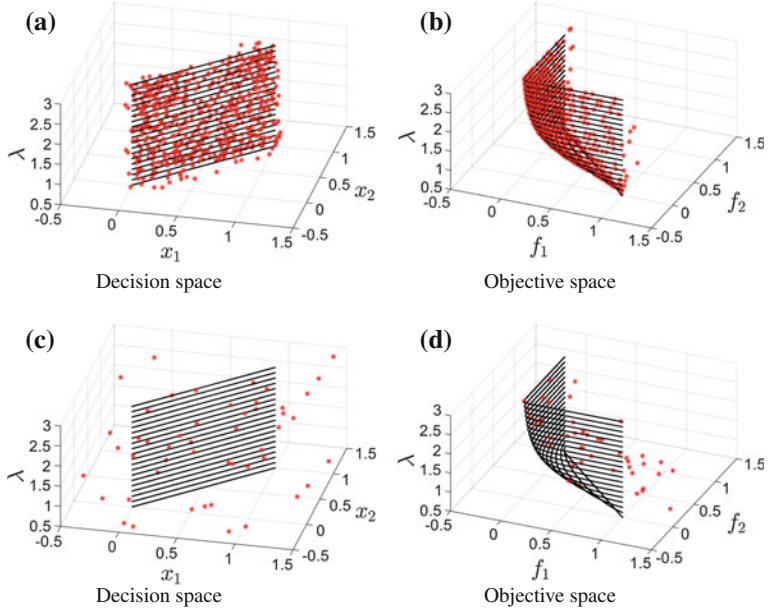


Fig. 28 Results for SNS (*above*) and GS (*below*) for PMOP4

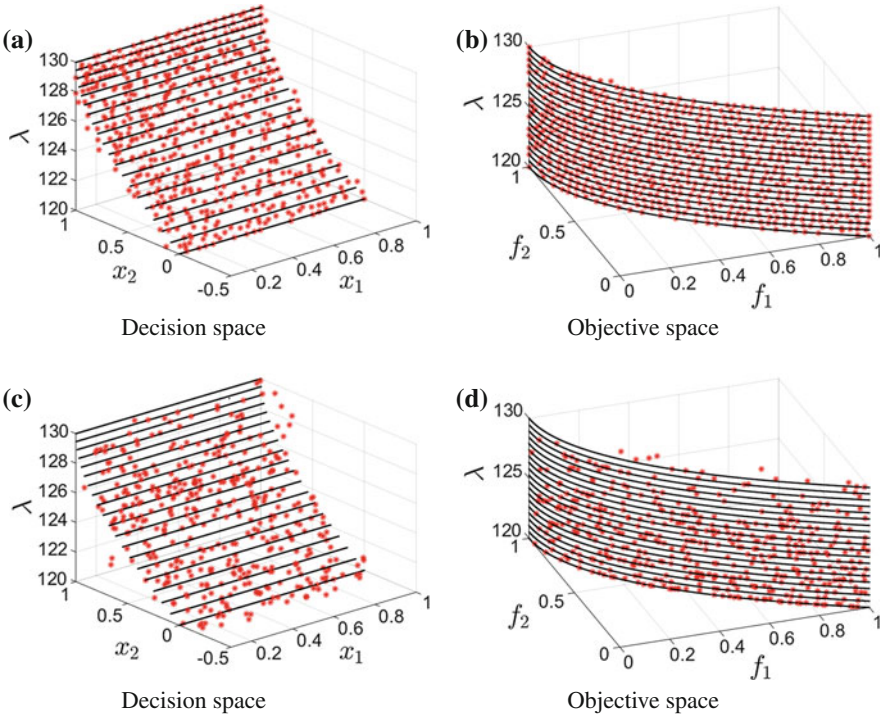


Fig. 29 Results for SNS (*above*) and GS (*below*) for PMOP5

Thus, though SNS is only based on SLS and has omitted all swarm-like strategies, one can say that it is already capable of computing a suitable finite size representation of the entire solution set within a reasonable amount of function evaluations. One can hence say that the problem of computing a finite size representation of $P_{Q,\Lambda}$ respectively $F(P_{Q,\Lambda})$ via stochastic search algorithms is well-posed.

6 Conclusions and Future Work

In this chapter, we have adapted and extended the Directed Search (DS) for the treatment of multi-objective optimization problems (MOPs) to the context of parameter dependent MOPs (PMOPs). Similar to DS, the new algorithm, λ-DS, is a point-wise iterative local search method that is able to steer the search into any direction d in objective space. After having explained the basic idea of the method including the discussion of the according greedy search in decision space, we have investigated a novel descent method (d is a descent direction, i.e., $d \leq 0$ and $d \neq 0$) as well as a continuation method (d points along the linearized family of Pareto fronts) that is capable of steering the search toward and along the set of interest, respectively. Next,

we have discussed how the method can be made gradient free via utilizing the existing neighborhood information. This feature makes the λ-DS a natural candidate for a local search engine within specialized evolutionary algorithms. Finally, we have used the approach to explain the behavior of stochastic local search (SLS) within PMOPs. Theoretical and empirical results indicate that both pressure toward and along the set of interest is already inherent in SLS (the former, however, by fixing the direction in λ space) which is strongly related to the terms convergence and spread as used in the evolutionary multi-objective optimization (EMO) community. Further, it can be seen that for points that are neither far away nor near to the solution set, SLS may find solutions in all directions in objective space which in principle allows SLS to detect solutions on all connected components of the solution set. Thus, one can say that the problem to find an approximation of the entire set of interest of a given PMOP via stochastic search algorithms is well-posed. We conjecture that these insights will have an impact on the future design of specialized evolutionary algorithms. One such example may be the design of novel local search engines that make use of the steering properties (e.g., adapted to constrained problems). Another observation might be that PMOPs do not have to be computed successively 'in slices', but that it may be advantageous to compute the entire solution set in one run of the algorithm.

Though the results presented in this chapter are already very promising, there are many aspects that are interesting for future work. For instance, all the considerations are done so far for unconstrained PMOPs, and an extension to constrained problems would be interesting. Next, the computations and comparisons have to be performed on more problems with different level of complexity. The integration of λ-DS into specialized evolutionary algorithms to design novel fast and reliable algorithms seems to be another promising task, in particular in the context of the gradient free realization of the local search method. We conjecture that the insights about the behavior of SLS might be helpful in the design of novel mutation operators. Finally, it might be interesting to apply λ-DS or a related hybrid algorithm to particular (real-world) online optimization problems.

Acknowledgments A. Sosa acknowledges support from the Conacyt to pursue his Ph.D. studies at the CINVESTAV-IPN. A. Lara acknowledges support from project SIP20162103. H. Trautmann acknowledges support from the European Center of Information Systems (ERCIS). All authors acknowledge support from DAAD project no. 57065955.

References

1. Allgower, E.L., Georg, K.: Numerical Continuation Methods. Springer, Heidelberg (1990)
2. Bank, B., Guddat, J., Klatte, D., Kummer, B., Tammer, K.: Non-Linear Parametric Optimization. Akademie, Berlin (1982)
3. Bosman, P.A.N.: On gradients and hybrid evolutionary algorithms for real-valued multiobjective optimization. IEEE Trans. Evol. Comput. **16**(1), 51–69 (2012)
4. Branke, J.: Evolutionary approaches to dynamic optimization problems-updated survey. In: GECCO Workshop on Evolutionary Algorithms for Dynamic Optimization Problems, pp. 27–30 (2001)

5. Branke, J., Kaußler, T., Schmidt, C., Schmeck, H.: A multi-population approach to dynamic optimization problems. In: Evolutionary Design and Manufacture, pp. 299–307. Springer, Heidelberg (2000)
6. Brown, M., Smith, R.E.: Signals directed multi-objective optimization. Int. J. Comput. Syst. **6**(1), 3–17 (2005)
7. Bu, Z., Zheng, B.: Perspectives in dynamic optimization evolutionary algorithm. In: Advances in Computation and Intelligence, pp. 338–348. Springer, Heidelberg (2010)
8. Carlos, A., Coello, C., Cortés, N.C.: Solving multiobjective optimization problems using an artificial immune system. Genet. Programm. Evolv. Mach. **6**(2), 163–190 (2005)
9. Das, I., Dennis, J.: Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems. SIAM J. Optim. **8**, 631–657 (1998)
10. Dellnitz, M., Schütze, O., Hestermeyer, T.: Covering Pareto sets by multilevel subdivision techniques. J. Optim. Theory Appl. **124**, (2005)
11. Dellnitz, Michael, Witting, Katrin: Computation of robust Pareto points. Int. J. Comput. Sci. Math. **2**(3), 243–266 (2009)
12. Ehrgott, M., Wiecek, M.M.: Multiobjective programming. In: Figueira, J., Greco, S., Ehrgott, M. (eds.) Multiple Criteria Decision Analysis: State of the Art Surveys, pp. 667–722. Springer, Heidelberg (2005)
13. Farina, M., Deb, K., Amato, P.: Dynamic multiobjective optimization problems: test cases, approximations, and applications. IEEE Trans. Evol. Comput. **8**(5), 425–442 (2004)
14. Gass, S., Saaty, T.: The computational algorithm for the parametric objective function. Nav. Res. Logist. Quart. **2**, 39–45 (1955)
15. Heinonen, J.: Lectures on Analysis on Metric Spaces. Springer, New York (2001)
16. Hernández, V.A.S.: On the Numerical Treatment of Parametric Multi-Objective Optimization Problems and Memetic Evolutionary Algorithms. Dissertation, CINVESTAV-IPN (2013)
17. Hillermeier, C.: Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach, vol. 135. Springer Science and Business Media, Heidelberg (2001)
18. Karush, W.E.: Minima of functions of several variables with inequalities as side conditions. Ph.D. thesis, University of Chicago (1939)
19. Kuhn, H., Tucker, A.: Nonlinear programming. In: Neumann, J. (ed.) Proceeding of the 2nd Berkeley Symposium on Mathematical Statistics and Probability, pp. 481–492 (1951)
20. Martín, A., Schütze, O.: A new predictor corrector variant for unconstrained bi-objective optimization problems. In: Tantar, A. et al., (ed) EVOLVE V, pp. 165–179. Springer, Heidelberg (2014)
21. Martin, B., Goldsztejn, A., Granvilliers, L., Jermann, C.: Certified parallelotope continuation for one-manifolds. SIAM J. Numer. Anal. **51**(6), 3373–3401 (2013)
22. Martin, B., Goldsztejn, A., Granvilliers, L., Jermann, C.: On continuation methods for nonlinear bi-objective optimization: towards a certified interval-based approach. J. Glob. Optim. 1–14 (2014)
23. Mejia, E., Schütze, O.: A predictor corrector method for the computation of boundary points of a multi-objective optimization problem. In: International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE 2010), pp. 395–399 (2010)
24. Miettinen, Kaisa: Nonlinear Multiobjective Optimization. Kluwer Academic Publishers, Boston (1999)
25. Nocedal, J., Wright, S.: Numerical Optimization. Springer Series in Operations Research and Financial Engineering, Springer, Heidelberg (2006)
26. Pareto, V.: Manual of Political Economy. The MacMillan Press, London (1971). (original edition in French in 1927)
27. Pereyra, V.: Fast computation of equispaced Pareto manifolds and Pareto fronts for multiobjective optimization problems. Math. Comput. Simul. **79**(6), 1935–1947 (2009)
28. Recchioni, M.C.: A path following method for box-constrained multiobjective optimization with applications to goal programming problems. Math. Methods Oper. Res. **58**, 69–85 (2003)
29. Ringkamp, M., Ober-Blöbaum, S., Dellnitz, M., Schütze, O.: Handling high dimensional problems with multi-objective continuation methods via successive approximation of the tangent space. Eng. Optim. **44**(6), 1117–1146 (2012)

30. Rudolph, G., Schütze, O., Grimme, C., Dominguez-Medina, C., Trautmann, H.: Optimal averaged hausdorff archives for bi-objective problems: theoretical and numerical results. Theoretical and Numerical Results Computational and Applied Optimization (to appear) (2016)
31. Schütze, O.: Set Oriented Methods for Global Optimization. Ph.D. thesis, University of Paderborn (2004). http://ubdata.uni-paderborn.de/ediss/17/2004/schuetze/
32. Schütze, O., Esquivel, X., Lara, A., Carlos, A., Coello, C.: Using the averaged Hausdorff distance as a performance measure in evolutionary multi-objective optimization. IEEE Trans. Evol. Comput. **16**(4), 504–522 (2012)
33. Schütze, O., Laumanns, M., Tantar, E., Carlos, C., Talbi, E.: Computing gap free pareto front approximations with stochastic search algorithms. Evol. Comput. **18**(1), 65–96 (2010)
34. Schütze, O., Martín, A., Lara, A., Alvarado, S., Salinas, E., Coello, C.A.C.: The directed search method for multi-objective memetic algorithms. Comput. Optim. Appl. 1–28 (2015)
35. Tantar, A-A., Danoy, G., Bouvry, P., Khan, S.U.: Energy-efficient computing using agent-based multi-objective dynamic optimization. In: Green IT: Technologies and Applications, pp. 267–287. Springer, Heidelberg (2011)
36. Tantar, A-A., Tantar, E., Bouvry, P.: A classification of dynamic multi-objective optimization problems. In: Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation, pp. 105–106. ACM, New York (2011)
37. Tinós, R., Yang, S.: A self-organizing random immigrants genetic algorithm for dynamic optimization problems. Genet. Program. Evolv. Mach. **8**(3), 255–286 (2007)
38. Van Veldhuizen, D.A.: Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations. Ph.D. thesis, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio (1999)
39. Witting, K.: Numerical Algorithms for the Treatment of Parametric Multiobjective Optimization Problems and Applications. Dissertation, Universität Paderborn (2012)
40. Witting, K., Ober-Blöbaum, S., Dellnitz, M.: A variational approach to define robustness for parametric multiobjective optimization problems. J. Glob. Optim. **57**(2), 331–345 (2013)
41. Witting, K., Schulz, B., Dellnitz, M., Böcker, J., Fröhleke, N.: A new approach for online multiobjective optimization of mechatronic systems. Int. J. Softw. Tools Technol. Trans. **10**(3), 223–231 (2008)
42. Yang, S.: Genetic algorithms with elitism-based immigrants for changing optimization problems. In: Applications of Evolutionary Computing, pp. 627–636. Springer, Heidelberg (2007)
43. Zhou, A., Jin, Y., Zhang, Q., Sendhoff, B., Tsang, E.: Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) Evolutionary Multi-Criterion Optimization. Lecture Notes in Computer Science, vol. 4403, pp. 832–846. Springer, Heidelberg (2007)