Chapter 5
Set Oriented Methods for the Numerical Treatment of Multiobjective Optimization Problems

Oliver Schütze, Katrin Witting, Sina Ober-Blöbaum, and Michael Dellnitz

Abstract. In many applications, it is required to optimize several conflicting objectives concurrently leading to a multobjective optimization problem (MOP). The solution set of a MOP, the Pareto set, typically forms a $(k-1)$-dimensional object, where k is the number of objectives involved in the optimization problem. The purpose of this chapter is to give an overview of recently developed set oriented techniques – subdivision and continuation methods – for the computation of Pareto sets $\mathscr{P}$ of a given MOP. All these methods have in common that they create sequences of box collections which aim for a tight covering of $\mathscr{P}$. Further, we present a class of multiobjective optimal control problems which can be efficiently handled by the set oriented continuation methods using a transformation into high-dimensional MOPs. We illustrate all the methods on both academic and real world examples.

Keywords: multiobjective optimization, multiobjective optimal control, set oriented methods, subdivision, continuation

5.1 Introduction

In a variety of applications one is faced with the problem that several objectives have to be optimized concurrently leading to a *multiobjective optimization problem* (MOP). Typically, the solution set of a MOP – the *Pareto set* – is not given by a single point as in scalar optimization but forms a $(k-1)$-dimensional object, where

Oliver Schütze
CINVESTAV-IPN, Computer Science Department, Av. IPN 2508, C. P. 07360,
Col. San Pedro Zacatenco Mexico City, Mexico
e-mail: schuetze@cs.cinvestav.mx

Katrin Witting · Sina Ober-Blöbaum · Michael Dellnitz
University of Paderborn, Chair of Applied Mathematics, Warburger Str. 100,
D-33098 Paderborn, Germany
e-mail: {witting,sinaob,dellnitz}@math.uni-paderborn.de

E. Tantar et al. (Eds.): EVOLVE- A Bridge between Probability, SCI 447, pp. 187–219.
springerlink.com

k is the number of objectives involved in the MOP. In case k is low (i. e., two to four), it makes sense to compute the entire solution set since this is the set of 'optimal compromises' and hence of particular interest for the decision making process.

In the literature, a huge variety of different methods for the computation of the Pareto set can be found. There exist, for instance, many scalarization methods which transform the MOP into a 'classical' scalar optimization problem (SOP). By choosing a clever sequence of SOPs a suitable finite size approximation of the entire Pareto set can be obtained (see [7, 40, 18, 32, 17, 16] and references therein). Further, there exist continuation methods that, starting from one or several solutions, perform a search along the Pareto set which is possible due to the geometry of the solution set (e. g., [26, 52, 23, 54]). Another way – and probably the most prominent one – is to use metaheuristics such as evolutionary algorithms (see [8, 20, 6] and references therein). The underlying idea is to evolve an entire set of solutions (population) during the optimization process. By this, an approximation of the entire Pareto set can be obtained by one single run of the algorithm.

The methods we consider here differ in the sense that in each iteration step a set of boxes is created with the aim to tightly cover $\mathscr{P}$. This can be done by subdivision techniques or by using certain continuation methods (so-called *recover techniques*). In the former, a sequence of nested box collections is generated that converges (ideally) to $\mathscr{P}$, and in the latter a given collection $\mathscr{C}$ is extended by a local search which is performed around promising elements (boxes) of $\mathscr{C}$. Subdivision techniques are due to their global approach highly competitive in particular if the dimension of the parameter space is moderate (say, $n < 50$), and the number of objectives is low ($k < 5$). Continuation methods are of local nature (i. e., restricted to the connected component of the solution set in which the given solution is contained), but in turn applicable to higher dimensional problems ($n \gg 1000$). Set oriented methods have been successfully applied to, for example, space mission design problems ([60, 13]), the design of electromagnetic shielding materials ([59]), the optimization of several subsystems of a rail-bound vehicle ([51, 37, 21, 22, 62, 56, 34]), an energy management problem of a tram ([33]), and the design of electrical circuits ([4]).

Next to the computation of the Pareto set of a given MOP we address the relatively young field of the numerical treatment of multiobjective optimal control problems. Whereas in multiobjective optimization one searches for Pareto optimal parameters, in optimal control one searches for optimal trajectories, which are solutions of a dynamical system given by a differential equation. Common approaches, such as *direct methods* (for an overview of different methods we refer to [3]), are based on a discretization of the trajectories and the differential equation such that in the end one is faced with a high-dimensional constrained (multiobjective) optimization problem. One of first works combining methods of direct optimal control and multiobjective optimization is e. g. [38]. In this contribution it is described how the set oriented continuation methods are combined with the recently developed direct optimal control method DMOC (Discrete Mechanics and Optimal Control [48]) which is in particular suitable for Lagrangian systems, e. g. systems in space mission design ([30, 42]), or constrained multi-body dynamics ([36, 49]). Additionally, the special case of differentially flat systems is addressed. We demonstrate on two examples that the resulting high-dimensional MOPs can be handled by the set oriented continuation methods.

The remainder of this chapter is organized as follows: In Section 5.2, we present the required background in multobjective optimization. In Section 5.3, we describe a subdivision technique for the computation of relative global attractors of a given dynamical system. In Section 5.4, we present four basic algorithms for the computation of Pareto sets, two subdivision algorithms and two continuation methods. In Section 5.5, we present two particular approaches for the treatment of multiobjective optimal control problems. And finally, in Section 10.6, we state some concluding remarks.

5.2 Multiobjective Optimization

In the following we consider MOPs which can be stated as follows:

$$\min_{x\in Q}\{F(x)\}, \quad Q=\{x\in\mathbb{R}^n : h(x)=0,\ g(x)\leq 0\}, \tag{5.1}$$

where F is defined as the vector of the objective functions, i. e.

$$F:Q\to\mathbb{R}^k, \quad F(x)=(f_1(x),\dots f_k(x)), \tag{5.2}$$

with $f_1,\dots,f_k : Q\to\mathbb{R}$, $h:Q\to\mathbb{R}^m$, $m\leq n$, and $g:Q\to\mathbb{R}^q$. Though the methods presented in the following are in principle applicable to general restriction sets Q, we will primarily consider unconstrained problems (i. e., $Q=\mathbb{R}^n$) or domains that result from box constraints, i. e.,

$$Q:=\{x\in\mathbb{R}^n\ :\ l_i\leq x_i\leq u_i,\ i=1,\dots,n\}, \tag{5.3}$$

where $l\in\mathbb{R}^n$ and $u\in\mathbb{R}^n$ define the lower and upper bounds, respectively.

In the next definition we state the classical concept of optimality for MOPs.

Definition 5.1. (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 $i\in\{1,\dots,k\}$. The relation $\leq_p$ is defined analogously.

(b) A vector $y\in\mathbb{R}^n$ is *dominated* by a vector $x\in\mathbb{R}^n$ (in short: $x\prec y$) with respect to (9.1) if $F(x)\leq_p F(y)$ and $F(x)\neq F(y)$.

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

In the following, we denote by P_Q the set of Pareto points (or Pareto set). The image $F(P_Q)$ of the Pareto set is called the *Pareto front*.

In case all the objectives are differentiable, the theorem of Kuhn and Tucker ([35]) states a necessary condition for optimality. We state the result in the following for the unconstrained case. For a more general formulation of the theorem we refer e. g. to [40].

Theorem 5.1 ([35]). *Let x^* be a Pareto point of (9.1). Then there exist vectors $\alpha \in \mathbb{R}^k$ with $\alpha_i \geq 0, i = 1, \ldots, k$, and $\sum_{i=1}^k \alpha_i = 1$ such that*

$$\sum_{i=1}^{k} \alpha_i \nabla f_i(x^*) = 0. \tag{5.4}$$

Points x^* that satisfy Equation (5.4) are called Karush-Kuhn Tucker[1] (KKT) points or substationary points. The above theorem can be used to give a qualitative description of P_Q (which has first been observed in [26]). Denote by $\tilde{F} : \mathbb{R}^{n+m+k} \rightarrow \mathbb{R}^{n+m+1}$ the following map:

$$\tilde{F}(x, \alpha) = \begin{pmatrix} \sum\limits_{i=1}^{k} \alpha_i \nabla f_i(x) \\ \sum\limits_{i=1}^{k} \alpha_i - 1 \end{pmatrix}. \tag{5.5}$$

By Theorem 5.1 it follows that for every KKT point $x^* \in \mathbb{R}^n$ there exists a vector $\alpha^* \in \mathbb{R}^k$ such that

$$\tilde{F}(x^*, \alpha^*) = 0. \tag{5.6}$$

Hence, one expects – as a result of the Implicit Function Theorem – that the set of KKT-points defines a $(k-1)$-dimensional manifold. This is indeed the case under certain smoothness assumptions, see [26] for a thorough discussion of this topic.

5.3 A Subdivision Algorithm for the Computation of Relative Global Attractors

The relative global attractor of a dynamical system contains all invariant sets and is hence (among other examples, see [11, 12]) interesting for the detection of substationary points of a given MOP. In the following we present the object of interest, the framework of a subdivision technique for the computation of such objects, and describe further on a numerical realization.

5.3.1 The Relative Global Attractor

Here we define the object of interest, the relative global attractor of a dynamical system. For a more detailled discussion we refer e. g. to [11, 12].

We consider discrete dynamical systems

$$\delta \int_0^{t_f} L(q(t), \dot{q}(t))\, dt + \int_0^{t_f} f(q(t), \dot{q}(t), u(t)))\, \delta q\, dt = 0 \tag{5.7}$$

[1] Named after the works of Karush [31] and Kuhn & Tucker [35] for scalar–valued optimization problems.

where $f : \mathbb{R}^n \to \mathbb{R}^n$. A subset $A \subset \mathbb{R}^n$ is called invariant if $f(A) = A$. We say an invariant set A is an attracting set if there exist a neighborhood U of A such that for every open set $V \supset A$ there is a $N \in \mathbb{N}$ such that $f^j(U) \subset V$ for all $j \geq N$. Note that for every invariant set also its closure is invariant. Hence, we can restrict ourselves to closed invariant sets A, and in this case we obtain

$$A = \bigcap_{j \in \mathbb{N}} f^j(U). \tag{5.8}$$

Hence, we can say that all the points in U are attracted by A (under iteration of f), and U is called the basin of attraction of A. If $U = \mathbb{R}^n$, then A is called the global attractor. The knowledge of the global attractor is in general beneficial since it contains all the potential interesting dynamics ([12]). For numerical aproximations, however, we have to restrict ourselves to a compact subset of the $\mathbb{R}^n$ as domain which leads directly to the notion of the relative global attractor.

Definition 5.2. Let $Q \subset \mathbb{R}^n$ be a compact set. The global attractor relative to Q is defined by

$$A_Q := \bigcap_{j \geq 0} f^j(Q). \tag{5.9}$$

Example 5.1. Consider the one-dimensional dynamical system

$$f(x) = \alpha x, \tag{5.10}$$

where $\alpha \in \mathbb{R}$ is a constant, and let $Q = [a,b]$, where $a < 0$ and $b > 0$.

(a) Let $\alpha \in (-1,1)$. Since $|x_{j+1}| = |\alpha||x_j|$ the relative global attractor is given by $A_Q = \{0\}$.
(b) Let $|\alpha| \geq 1$. Since for all $j \in \mathbb{N}$ it is $f^j(Q) \supset Q$ and $f^0(Q) = Q$, it is $A_Q = Q$.

As an example related to optimization consider the application of the steepest descent method ([45]) to a scalar optimization problem

$$\min_x g(x), \tag{5.11}$$

where $g : \mathbb{R}^n \to \mathbb{R}$ is a smooth function. This leads to the dynamical system

$$x_{j+1} = f(x_j) = x_j - t\nabla g(x_j), \quad j = 0,1,2,\ldots, \tag{5.12}$$

where $t \in \mathbb{R}_+$ is a (fixed) step size. It is important to note that the relative global attractor contains all invariant sets $A \subset Q$ and is hence interesting in the present context:

Let $x^* \in \mathbb{R}^n$ be a substationary point (i. e., $\nabla g(x^*) = 0$), then it is

$$x^* = f(x^*) \tag{5.13}$$

i. e., x^* is a fixed point of f (and in particular invariant). Note that this statement holds regardless of the choice of the step size t.

5.3.2 The Algorithm

Here we describe a subdivision technique that creates in each iteration step j a collection of sets Q_j such that each Q_j is an outer approximation of A_Q and that the sequence of Q_j's converges to A_Q in the Hausdorff sense.

Let $\mathscr{B}_0$ be an initial collection of finitely many subsets of the compact set Q such that $\cup_{B\in\mathscr{B}_0} B = Q$. Then $\mathscr{B}_j$ is inductively obtained from $\mathscr{B}_{j-1}$ in two steps:

(i) Subdivision Construct from $\mathscr{B}_{j-1}$ a new system $\hat{\mathscr{B}}_j$ of subsets such that

$$\bigcup_{B\in\hat{\mathscr{B}}_j} B = \bigcup_{B\in\mathscr{B}_{j-1}} B \tag{5.14}$$

and

$$\operatorname{diam}(\hat{\mathscr{B}}_j) = \theta_j \operatorname{diam}(\mathscr{B}_{j-1}), \tag{5.15}$$

where $0 < \theta_{min} \leq \theta_j \leq \theta_{max} < 1$.

(ii) Selection Define the new collection $\mathscr{B}_j$ by

$$\mathscr{B}_j = \left\{B \in \hat{\mathscr{B}}_j : \text{there exists } \hat{B} \in \hat{\mathscr{B}}_j \text{ such that } f^{-1}(B) \cap \hat{B} \neq \emptyset\right\}. \tag{5.16}$$

Denote by Q_j the collection of compact subsets obtained after j subdivision steps, i. e.,

$$Q_j := \bigcup_{B\in\mathscr{B}_j} B \tag{5.17}$$

One can show that the limit of the Q_j's converges to the relative global attractor.

Proposition 5.1 ([12]). *Let A_Q be a global attractor relative to the compact set Q, f be a diffeomorphism, and let $\mathscr{B}_0$ be a finite collection of closed subsets with $Q_0 := \cup_{B\in\mathscr{B}_0} B = Q$. Then*

$$A_Q = \bigcap_{j=0}^{\infty} Q_j. \tag{5.18}$$

The above result can alternatively be stated as

$$\lim_{j\to\infty} d_H(A_Q, Q_j) = 0, \tag{5.19}$$

where $d_H(\cdot,\cdot)$ denotes the Hausdorff distance between two sets.

Note that the above result holds for the usage of *one* dynamical system throughout the entire iteration process. In the context of optimization, however, this might be too restrictive. As a general example, consider the dynamical system (5.12).

Instead of using a fixed step size t, one is typically interested in using several step sizes which formally leads to an entire family of dynamical systems. In the case of steepest descent this would be

$$x_{j+1} = f_i(x_j) = x_j + t_i \nabla g(x_j), \quad i \in \mathscr{I} \tag{5.20}$$

For an adaption of the subdivision technique to that context we refer to [15].

5.3.3 Realization of the Algorithm

Here we describe a possible realization of the subdivision technique.

Subdivision

For the representation of the collections $\mathscr{B}_j$ we use boxes: Let us assume that every parameter is restricted to a certain range, i. e., $a_i \leq x_i \leq b_i$, $i = 1, \ldots, n$. The search space thus is given by

$$Q = [a_1, b_1] \times \ldots \times [a_n, b_n] \subset \mathbb{R}^n. \tag{5.21}$$

Every box $B \subset \mathbb{R}^n$ can be represented by a center $c \in \mathbb{R}^n$ and a radius $r \in \mathbb{R}^n_+$ such that

$$B = B^{(c,r)} = \{x \in \mathbb{R}^n \,:\, |x_i - c_i| \leq r_i \; \forall i = 1, \ldots, n\}. \tag{5.22}$$

The box B can be subdivided with respect to the j-th coordinate. This division leads to two boxes $B_-^{(c^-,\hat{r})}$ and $B_+^{(c^+,\hat{r})}$, where

$$\hat{r}_i = \begin{cases} r_i & \text{for } i \neq j \\ r_i/2 & \text{for } i = j \end{cases}, \quad c_i^{\pm} = \begin{cases} c_i & \text{for } i \neq j \\ c_i \pm r_i/2 & \text{for } i = j \end{cases}.$$

Let $P(Q,0) := Q$, that is, $P(Q,0) = B^{(c^0,r^0)}$, where

$$c_i^0 = \frac{a_i + b_i}{2}, \quad r_i^0 = \frac{b_i - a_i}{2}, \quad i = 1, \ldots, n.$$

Denote by $\mathscr{P}(Q,d), d \in \mathbb{N}$, the set of boxes obtained after d subdivision steps starting with $B^{(c^0,r^0)}$, where in each step $i = 1, \ldots, d$ the boxes are subdivided with respect to the j_i-th coordinate, where j_i is varied cyclically. That is, $j_i = ((i-1) \bmod n) + 1$. Note that for every point $y \in Q \backslash \partial Q$ and every subdivision step d there exists exactly one box $B = B(y,d) \in \mathscr{P}(Q,d)$ with center c and radius r such that $c_i - r_i \leq y_i < c_i + r_i$, $\forall i = 1, \ldots, n$. Thus, every set of solutions $\mathscr{S}_B$ leads to a set of box collections $\mathscr{B}_d$. These collections can easily be stored in a binary tree with depth d. In Figure 5.1 a representation of five boxes with subdivision step three and three dimensions (i. e., $n = 3$) together with the corresponding set $\mathscr{B}_3$ is shown. Note that each $\mathscr{B}_d$ is completely determined by the tree structure and the initial box $B^{(c^0,r^0)}$. Using this scheme, the memory requirements grow only linearly in the dimension n of the problem.

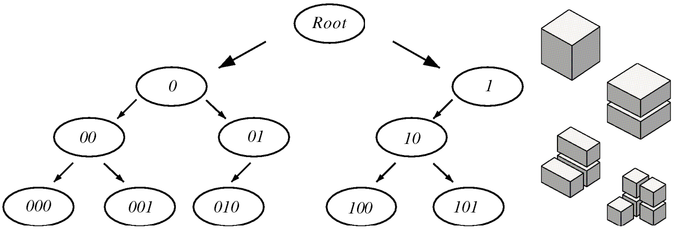


Fig. 5.1 The data structure used for the representation of the solution set

Selection

A box B is removed from the collection in the above algorithm if

$$\forall \hat{B} \in \mathscr{B}_k : f^{-1}(B) \cap \hat{B} = 0 \tag{5.23}$$

Apparently, this is hard to decide apart for trivial problems. As a remedy, the following heuristic can be chosen which has shown its efficiency in numerous examples: One can discretize each box of the collection by selecting a finite set of test points (e. g. , grid points in low dimensions n of the parameter space or Monte Carlo points in higher dimensions). Then, one can replace removal strategy (5.23) by the following one:

$$f(x) \notin B \quad \text{for all test points } x \in \hat{\mathscr{B}}_k. \tag{5.24}$$

Similar strategies can be found in cell-mapping techniques (e. g., [28, 29]).

5.4 Basic Algorithms for Multiobjective Optimization

In the following, we present four different algorithms for the numerical treatment of MOPs, two subdivision algorithms and two continuation strategies. In all cases, we emphasize on the general idea, for details or comparisons to other methods we refer to the original works.

5.4.1 Subdivision Techniques

5.4.1.1 DS-Subdivision

The first algorithm we present here is in principle constructed as the one presented in (5.12), albeit tailored to the context of multiobjective optimization ([14]). Assume the MOP is unconstrained and all objectives are continuously differentiable. The following result gives a way to compute a descent direction – i. e., a direction $v \in \mathbb{R}^n$ where all objectives can be improved simultaneously – at every non-optimal point $x \in \mathbb{R}^n$.

Theorem 5.2 ([55]). *Let (MOP) be given and $q : \mathbb{R}^n \to \mathbb{R}^n$ be defined by*

$$q(x) = \sum_{i=1}^{k} \hat{\alpha}_i \nabla f_i(x), \tag{5.25}$$

where $\hat{\alpha}$ is a solution of

$$\min_{\alpha \in \mathbb{R}^k} \left\{ \left\| \sum_{i=1}^{k} \alpha_i \nabla f_i(x) \right\|_2^2 ; \alpha_i \geq 0, i = 1, \ldots, k, \sum_{i=1}^{k} \alpha_i = 1 \right\}. \tag{5.26}$$

Then either $q(x) = 0$ or $-q(x)$ is a descent direction for all objective functions $f_1, \ldots, f_k$ in x.

Note that since each x with $q(x) = 0$ is a substationary point, the computation of the descent direction includes a test for Pareto optimality.

Having the descent direction q, a possible dynamical system that 'pushes' the iterates toward the set of interest, the Pareto set, is now at hand: Analog to the line search method in (5.12) we can define

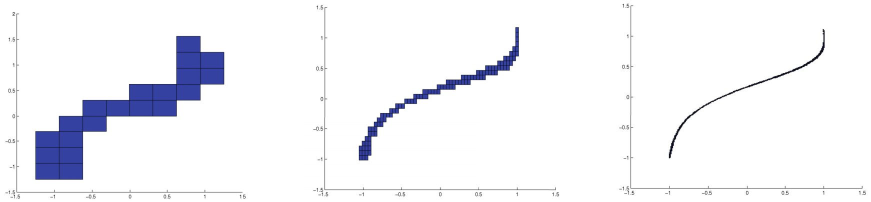
$$x_{j+1} = f(x_j) = x_j - tq(x_j), \quad j = 0, 1, 2, \ldots, \tag{5.27}$$

where $t \in \mathbb{R}_+$ is a chosen step size. DS-Subdivision is the subdivision technique described in Section 5.3.2 using (5.27) as dynamical system.

Example 5.2. Consider the following bi-objective problem

$$\begin{aligned} f_1, f_2 : & \mathbb{R}^2 \to \mathbb{R} \\ f_1(x) &= (x_1 - 1)^4 + (x_2 - 1)^2, \\ f_2(x) &= (x_1 + 1)^2 + (x_2 + 1)^2 \end{aligned} \tag{5.28}$$

The Pareto set of MOP (5.28) is a curve connecting the points $(-1,-1)^T$ and $(1,1)^T$. Figure 5.2 shows the result of an application of the subdivision scheme where (5.27) has been used as dynamical system. After several iteration steps a tight covering of the Pareto set can be obtained. For the evaluation of a box we have chosen the four corners as test points.



(a) Iteration 10 (b) Iteration 15 (c) Iteration 20

Fig. 5.2 Box collections generated by DS-Subdivision applied on MOP (5.28) after 10, 15, and 20 iteration steps. Here, we have chosen $Q = [-5,5]^2$ as domain.

Indeed one can show convergence to the set of interest if it is connected.

Proposition 5.2 ([14]). *Suppose that the set $\mathscr{S}$ of substationary points is bounded and connected. Let Q be a compact neighborhood of $\mathscr{S}$. Then, an application of the subdivision algorithm to Q with respect to the iteration scheme (5.27) leads to a sequence of coverings which converges to the entire set $\mathscr{S}$; that is,*

$$d_H(\mathscr{S}, Q_j) \to 0, \quad \textit{for } j \to \infty. \tag{5.29}$$

If e. g. the problem is convex, then it is known that $\mathscr{S}$ is equal to P_Q which is furthermore connected. Unfortunately, analog results cannot be obtained for the case where the set $\mathscr{S}$ is disconnected. The reason for this is that the relative global attractor is always connected. The following example demonstrates this in the present context.

Example 5.3. Consider the bi-objective problem as shown in Figure 5.3. This problem is constructed such that $\mathscr{S} = [0,1] \cup [1.5,2]$, where the interval $[0,1]$ contains only locally optimal solutions and the interval $[1.5,2]$ is equal to the Pareto set. An application of DS-Subdivision to $Q = [-1,3]$ will converge to the relative global attractor $A_Q = [0,2]$. To see this, one has to consider the neighborhood around the number 1: a box B that contains 1 as well as points that are bigger than one has always a nonzero intersection with its image under iteration (5.27). Further, the image of this box has also a nonzero intersection with its right neighbor B_r. Proceeding with B_r, we see that all the boxes between 1 and 1.5 have preimages in other boxes in each step of the subdivision process. Hence, the interval (1,1.5) is never removed in the selection step.

However, it has to be noted that this holds for an ideal application of the algorithm. In case the removal strategy (5.24) is used in the selection strategy, it is most likely to observe convergence toward $\mathscr{S}$.

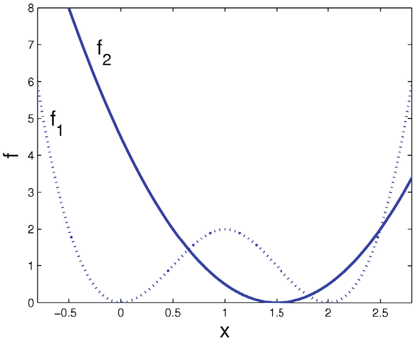


Fig. 5.3 Example of a bi-objective optimization problem where the set $\mathscr{S}$ of locally optimal solutions is disconnected

Remark 5.1. We have utilized in our studies the descent method presented in [55]. However, we have to note that there are other ways to compute descent directions (e. g., [18, 5]) which might lead to similar results.

5.4.1.2 Sampling Algorithm

Note that the algorithm described above suffers several potential drawbacks, namely:

(a) The objectives' gradients have to be at hand or have to be approximated.
(b) The set $\mathscr{S}$ of substationary points is typically a strict superset of the Pareto set, and points that are only locally optimal are typically not of interest (compare to Example 5.3).
(c) The algorithm is in principle capable of finding local Pareto points on the boundary of the domain Q. However, empirical tests have shown that in many cases a significant fraction of the boundary ∂Q is locally but not globally optimal wrt the given MOP.

The following algorithm, the *Sampling Algorithm*, tries to avoid all these potential problems. This is done by merely considering the objective values of the set of test points in each iteration. To be more precise, given a box collection $\mathscr{B}_{j-1}$, the collecion $\mathscr{B}_j$ is obtained as follows:

(i) Subdivision This is as in Section 5.3.2.
(ii) Selection For all $B \in \hat{\mathscr{B}}_j$, choose a set of test points $X_B \subset B$.
$N_j :=$ nondominated points of $\bigcup_{B\in\hat{\mathscr{B}}_j} X_B$
$\mathscr{B}_j := \{B \in \hat{\mathscr{B}}_j \; : \; \exists y \in X_B \cap N_j\}$

Note that this approach has some analogies to branch and bound strategies used for scalar optimization problems (e. g., [27]), but omits any bounding strategy. This is due to the fact that the larger the number k of objectives is, the more robust the selection strategy gets (note that for $k = 1$, N_j will typically consist of one element, which is normally not the case for $k > 1$).

Example 5.4. Consider the following bi-objective problem taken from [55]:

$$\begin{aligned} &f_1, f_2 : \mathbb{R}^n \to \mathbb{R}, \\ &f_1(x) = \sum_{j=1}^{n} x_j, \\ &f_2(x) = 1 - \prod_{j=1}^{n} (1 - w_j(x_j)), \end{aligned} \tag{5.30}$$

where

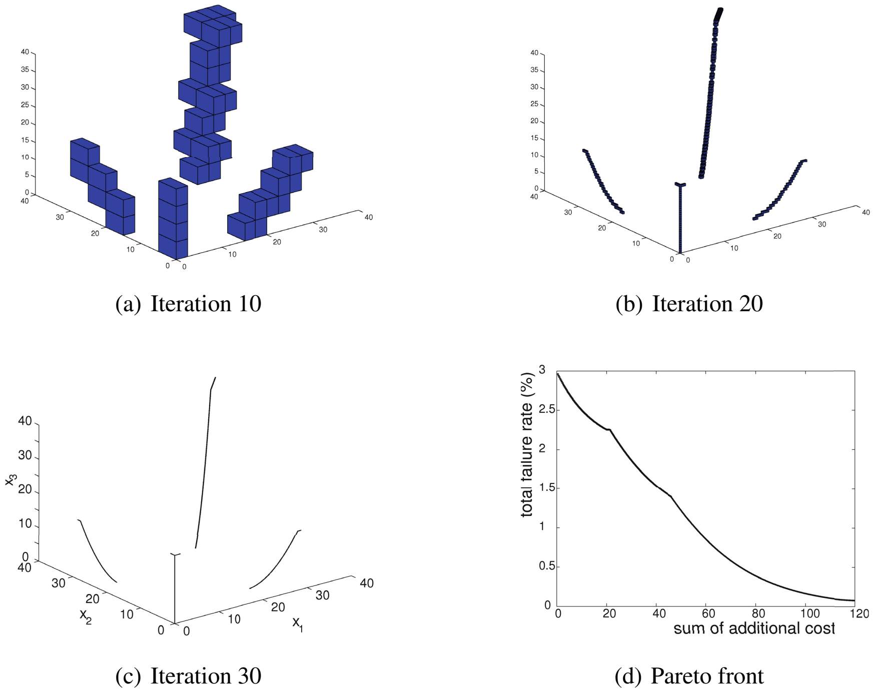
$$w_j(z) = \begin{cases} 0.01 \cdot \exp(-(\frac{z}{20})^{2.5}) & \text{for} \quad j = 1,2 \\ 0.01 \cdot \exp(-\frac{z}{15}) & \text{for} \quad 3 \leq j \leq n \end{cases}$$

Figure 5.4 shows a numerical result obtained by the Sampling Algorithm. Here, we have taken 10 randomly chosen test points per box. When choosing $Q = [0,40]^3$ the set $\mathscr{S}$ contains the two faces of Q with $x_i = 0$, $i = 1,2$. Hence, an application of DS-Subdivision leads to a tremendous effort since both faces will be kept in the box collections (see also [57]). This is avoided by the sampling approach.

5.4.2 Recover Techniques in Parameter Space

In the course of the two algorithms described above it can happen that boxes are lost that contain a part of the set of interest (e. g., due to a discretization error in the removal strategy (5.24)). The following algorithms are intended to 'heal' (or recover) the box collection. The underlying idea is that the set of interest (Pareto set or front) forms locally a manifold. That is, in the neighborhood of a 'good' box (i. e., a box with nonzero intersection with the set of interest) it is likely that there are other 'good' boxes due to the geometry of the problem. Hence, given a box collection $\mathscr{B}_j$, it makes sense to perform a local search around each box of the collection (once), and to see if neighboring boxes should be added to $\mathscr{B}_j$. It has to be noted that this approach is restricted to the connected components of the set of interest that have nonzero intersection with the given collection $\mathscr{B}_j$. On the other hand, it has turned out that the usage of the data structure is well suited to maintain a 'global' view on the part of the solution set which is already computed, and is in particular interesting for the efficient treatment of high-dimensional problems.

The idea of the recover techniques in parameter space is to recover the box collection in order to maintain a perfect covering of the Pareto set ([14]). The following



(a) Iteration 10

(b) Iteration 20

(c) Iteration 30

(d) Pareto front

Fig. 5.4 Numerical results for MOP (5.30) using the Sampling Algorithm.

pseudo-code gives the framework of the Recover Algorithm to extend an existing collection $\mathscr{B}_j$ (see also Figure 5.5).

```
(i) Step 1    Mark all boxes $B \in \mathscr{B}_j$
(ii) Step 2   (i) For all marked $B \in \mathscr{B}_k$: unmark the box and choose starting points
    $(s_i)_{i=1,\ldots,l}$ near $B$
    (ii) For each $s_i$, $i = 1,\ldots,l$, compute a substationary point $p_i$ starting from $s_i$.
    (iii) For all $p_i$, $i = 1,\ldots,l$, if $B(y,j) \notin \mathscr{B}_k$, add $B(y,j)$ to the collection $\mathscr{B}_k$ and
    mark the box.
    (iv) Repeat Step 2 while new boxes are added to $\mathscr{B}_k$ or until a prescribed number
    of steps is reached.
```

Note that the Recover Algorithm is similar in spirit to predictor corrector (PC) methods used for numerical (multiobjective) continuation ([53, 1, 26, 24, 52, 23]). Crucial are certainly the proper choices of the starting points s_i and the performance of the local searcher. In low dimensions it might be sufficient to choose the starting points in coordinate directions from the center of a box (as seen in Figure 5.5) together with an application of the map (5.27), i. e., to take $p_i = f^p(s_i)$, for a power $p \in \mathbb{N}$. In higher dimensions, however, this is not advisable. Instead, it makes sense to lean elements from existing PC methods applied on the map (9.11). This has been done in [58]. For an application of the recover techniques for high-dimensional problems ($n \gg 1000$) we refer to [54].

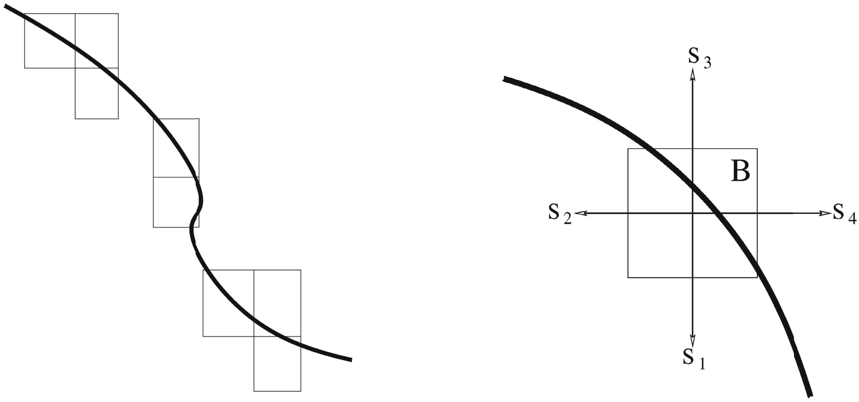


Fig. 5.5 Recover algorithm: uncomplete covering of the Pareto set (left) and possible choice of test points for a given box B (right)

Example 5.5. We consider the following MOP ([58]):

$$\begin{aligned} \min F(x) &:= \begin{pmatrix} (x_1-1)^4+(x_2-1)^2+(x_3-1)^2 \\ (x_1+1)^2+(x_2+1)^4+(x_3+1)^2 \\ (x_1-1)^2+(x_2+1)^2+(x_3-1)^4 \end{pmatrix} \\ \text{s.t. } h(x) &= 1-x_3^2-(\sqrt{x_1^2+x_2^2}-4)^2=0 \end{aligned} \tag{5.31}$$

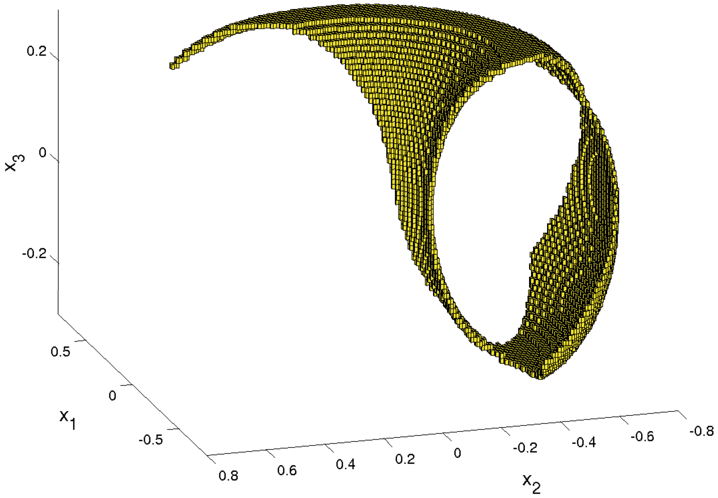
The MOP is given by three convex objectives which are constrained to a torus. Figure 5.6 shows a numerical result of the Recover Algorithm where the initial box collection consists of one single solution of the MOP.

We stress that the Recover Algorithm can more generally be used as a particular continuation method for the numerical solution of

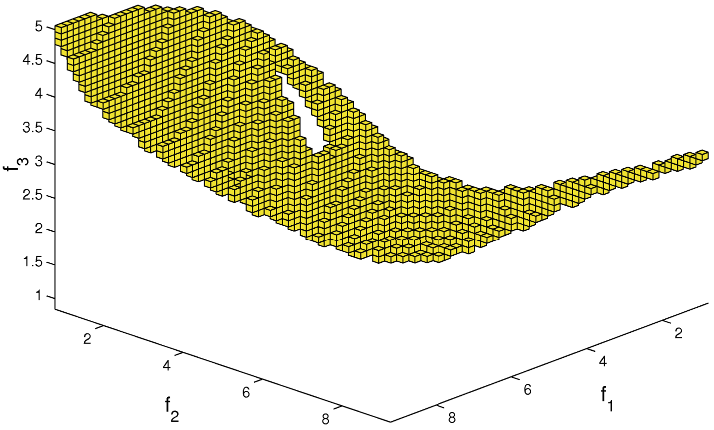
$$H(x)=0, \tag{5.32}$$

where $H : \mathbb{R}^{N+K} \to \mathbb{R}^N$ is a map (see [57, 58]). One interesting application in the present context is the numerical treatment of parameter dependent MOPs which can be expressed as follows:

$$\min_x F_\lambda : \mathbb{R}^n \to \mathbb{R}^k, \qquad \lambda \in \mathbb{R}^d \tag{5.33}$$



(a) Box collection



(b) Pareto front

Fig. 5.6 Numerical result of the Recover Algorithm for MOP (5.31) starting with one known Pareto optimal solution of the problem

This particular kind of problem e. g. occurs when λ is given data for the underlying system which is modelled by F and can change during the optimization process (see e. g. [51]). In case λ changes quickly it is not advisable to compute the entire Pareto set for every value of λ but it may be more efficient to approximate the set $\tilde{F}^{-1}(0)$, where

$$\tilde{F} : \mathbb{R}^{n+d+k} \to \mathbb{R}^{n+1}$$

$$\tilde{F}(x,\lambda,\alpha) := \begin{pmatrix} \sum\limits_{i=1}^{k} \alpha_i \frac{\partial f_i}{\partial x}(x,\lambda) \\ \sum\limits_{i=1}^{k} \alpha_i - 1 \end{pmatrix}. \tag{5.34}$$

When the auxiliary system is computed, the set of substationary points for every value $\bar{\lambda}$ is given by the projection $\tilde{F}^{-1}(0)|_{\lambda=\bar{\lambda}}$, which can easily be identified in the corresponding box collection.

Example 5.6. We consider the following parameter dependent MOP:

$$F_\lambda(x) := (1-\lambda)F_1(x) + \lambda F_2(x), \tag{5.35}$$

where

$$F_1, F_2 : \mathbb{R}^2 \to \mathbb{R}^2$$

$$F_1(x_1,x_2) = \begin{pmatrix} (x_1-1)^4 + (x_2-1)^2 \\ (x_1+1)^2 + (x_2+1)^2 \end{pmatrix}, \tag{5.36}$$

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

Figure 5.7 shows the set $\tilde{F}^{-1}(0)$ for problem (5.35). Two 'classical' Pareto sets for particular values of λ – using the according parts of the box collection for the auxiliary system – can be seen in Figure 5.8.

5.4.3 Image-Set Oriented Recover Techniques

In case the dimension of the parameter space is high and only a few objectives are beeing considered (i. e., two or three), one can alternatively generate box collections in image space ([9, 10]): For a given initial Pareto optimal solution a box on the Pareto front is generated around the image of this solution. Step by step, all neighboring boxes are inserted that contain points on the Pareto front. The insertion of boxes is based on the idea to create vectors of desired values for the objectives,

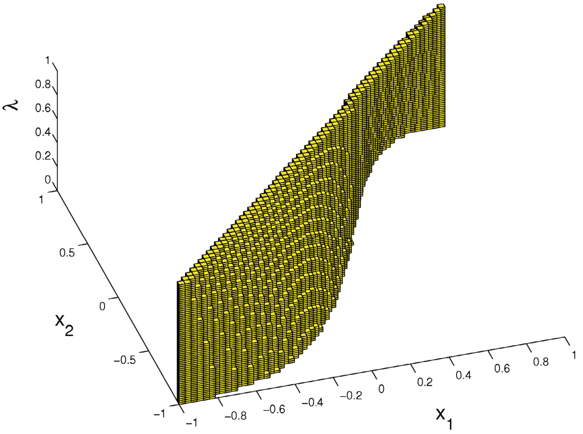
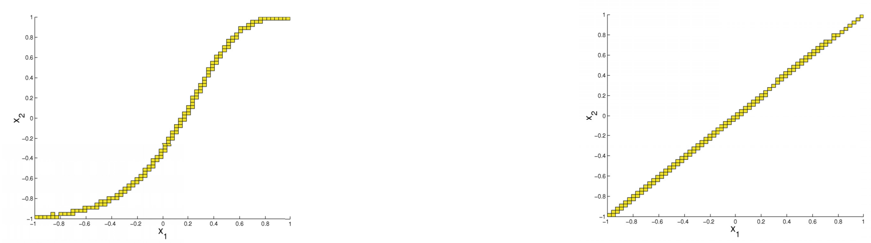


Fig. 5.7 Family of Pareto sets, see (5.35)



(a) $\lambda = 0$

(b) $\lambda = 1$

Fig. 5.8 Pareto sets for two values of λ of MOP (5.35)

so-called *targets* T, in the neighborhood of the given boxes. Then the following distance minimization problem is solved:

$$\min_{x} \|F(x) - T\|_2. \tag{5.37}$$

Using this procedure, the entire Pareto set can be covered for unconstrained multi-objective optimization problems with convex objective functions. In the nonconvex case the connected components of the Pareto front that correspond to the initial boxes can be approximated.

More precisely, the image-set oriented recover algorithm works as follows: Assume that we would like to compute Pareto optimal points within the region

$$Q_I = [f_1^{\min}, f_1^{\max}] \times \ldots \times [f_k^{\min}, f_k^{\max}] \subset \mathbb{R}^k,$$

where $f_i^{\min}, f_i^{\max} \in \mathbb{R}$, $i = 1, \dots, k$, are given restrictions for the objective values. The box Q_I is subdivided a set of boxes of depth d, $\mathscr{P}(Q_I, d)$, as described in Section 5.3. Then, each point $y \in Q_I$ can be assigned to a box $B(y, d)$.

The image-set oriented recover algorithm starts with a box collection $\mathscr{B}_0 \subset \mathscr{P}(Q_I, d)$. Let x_B denote a corresponding Pareto optimal solution to the box B, i. e. $F(x_B) \in B$ for $B \in \mathscr{B}_0$.

```
Step 1:
        Mark all $B \in \mathscr{B}_0$
Step 2:
        for $j = 0, \dots,$ maximum number of steps:
            set $\hat{\mathscr{B}}_j = \mathscr{B}_j$
                for all $B \in \mathscr{B}_j$ with $B$ marked
                    choose target vectors $\{T_i\}_{i=1,\dots,l}$ near $B$
                        with $T_i \leq_p F(x_B)$
                    compute $x_i^\star = \arg\min_{x \in \mathbb{R}^n} \|F(x) - T_i\|_2$ for $i = 1, \dots, l$,
                    set $F_i^\star = F(x_i^\star), i = 1, \dots, l$,
                    unmark box $B$
                    for all $i = 1, \dots, l$:
                        if $B(F_i^\star, d) \notin \mathscr{B}_j$
                            set $\tilde{\mathscr{B}} = B(F_i^\star, d)$, $x_{\tilde{B}} = x_i^\star$, $F_{\tilde{B}} = F_i^\star$
                            mark $\tilde{B}$
                            $\hat{\mathscr{B}}_j = \hat{\mathscr{B}}_j \cup \tilde{B}$
                if $\hat{\mathscr{B}}_j == \mathscr{B}_j$ STOP
                $\mathscr{B}_{j+1} = \hat{\mathscr{B}}_j$
```

So far, it has not been explained how suitable targets T_i can be generated. Efficient strategies for the computation of target vectors can be defined by making use of local information on the Pareto set. There are different possibilities how to generate good targets. One idea presented in [10] is to generate targets along the shifted tangent space on the image of a known Pareto optimal solution. More precisely, we have to assume that $x^\star$ is Pareto optimal, $F(x^\star) = y^\star$ and $T^\star$ is the target which leads to the computation of $x^\star$. Additionally, it is required that the image of the Pareto set is smooth and forms a $(k-1)$-dimensional manifold in a neighborhood of $y^\star$. Then, new targets can be generated in two steps:

(i) Compute the normal vector to the Pareto front in the point $y^\star$. As $x^\star$ is a solution of the distance minimization problem (5.37) this normal vector is given by $n = \frac{T^\star - x^\star}{\|T^\star - x^\star\|}$. Construct a (($k$-1)-dimensional) orthonormal basis $V = \{b_1, \dots, b_{k-1}\}$ of the tangent space at the point $y^\star$ which is orthogonal to n e. g. by computing a QR factorization of n.

(ii)Specify l targets

$$t_i = y_i^\star + \sum_{j=1}^{k-1} \alpha_{i,j} b_j + \lambda_i n, \quad i = 1, \dots, l.$$

The coefficients $\alpha_{i,j}$ are chosen in such a way that the points $p_i = \sum_{j=1}^{k-1} \alpha_{i,j} b_j$ are located inside neighboring boxes of the box containing $y^\star$. The value of λ_i is determined by an adaptive concept which guarantees that the targets lie below the Pareto front (but also are not too far away).

The distance minimization problem (5.37) is solved using standard optimization algorithms such as SQP which is implemented in the NAG library [46]. In Figure 5.9, a schematic representation of the image set-oriented recover algorithm is given.

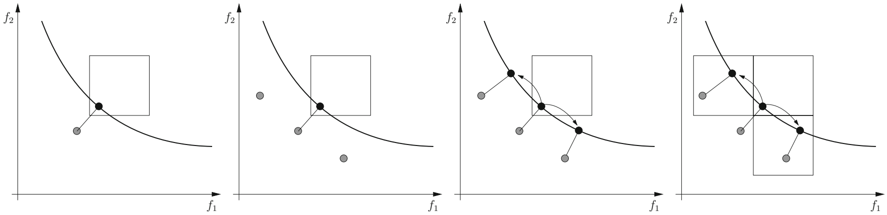


Fig. 5.9 Principal functioning of the image set-oriented recover algorithm (the black curve is the unknown Pareto front and the grey dots are the targets).

Example 5.7. Consider the bi-objective problem

$$\begin{aligned} &f_1, f_2 : \mathbb{R}^{100} \to \mathbb{R}, \\ &f_1(x) = \sum_{i=1}^{100} (x_i - 1)^2 \\ &f_2(x) = \sum_{i=1}^{100} (x_i + 1)^2. \end{aligned} \tag{5.38}$$

We restrict the optimization to a box with center $(0,0)^T$ and radius 2 in each spatial direction in image space. To demonstrate the application of the image set oriented recover algorithm this box is subdivided into boxes of depth 12 in a first study, and depth 18 in a second study. As a start, we consider the box of depth 12 or 18, respectively, containing the point $(0.25, 0.25)^T$ which lies on the Pareto front (this point can for example be computed by minimizing the weighted sum $\frac{1}{2} f_1(x) + \frac{1}{2} f_2(x)$). In Figure 5.10 the results from the application of the image-set oriented recover algorithm to this example is given for the two different box depths mentioned above. One can observe that the entire Pareto front is covered by boxes of the respective depth.

Example 5.8. The image-set oriented recover algorithm has been applied to an energy management problem of a tram which is supplied by an overhead line (cf. [9, 33]). This tram posesses an additional onboard storage system with an energy storage of high capacity which is able to store energy generated from breaking, for

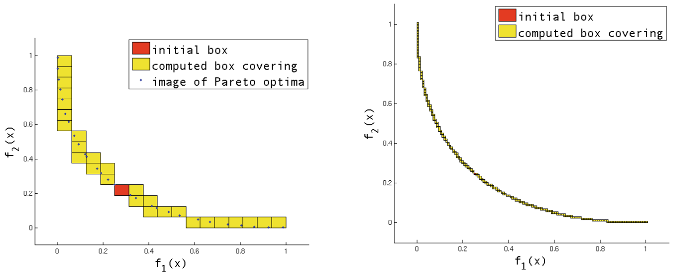


Fig. 5.10 Results of the image-set oriented recover algorithm applied to the objective functions given in Example 5.7: Box covering of the Pareto front for depth $d = 12$ (left) and $d = 18$ (right)

example. The aim is to reduce both overhead line peak power and energy consumption simultaneously during a realistic drive cycle of the tram. To compute reasonable solutions the drive cycle under consideration is divided into 1241 track sections. The energy management system has to assign a reference value to each of these sections. Thus, a multiobjective optimization problem with two objectives and 1241 optimization parameters has to be solved. Figure 5.11 shows the results. Here, both objectives are normed in such a way that they each equal one if no energy storage system would be used. Note that in the figure not the resulting boxes in image space but the solutions within these boxes are plotted.

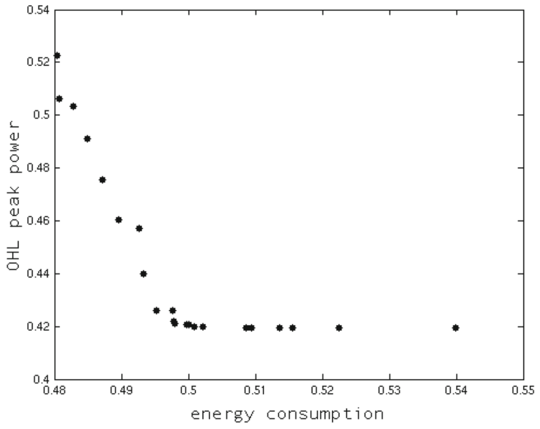


Fig. 5.11 Approximation of the Pareto front for the energy management problem of a tram described in Example 5.8

5.5 Multiobjective Optimal Control Problems

In this section we consider multiobjective optimal control problems of the form

$$\min_{x,u} J(x,u) \quad \text{s. t. } \dot{x}(t) = g(x(t),u(t)), \tag{5.39}$$

where $x : [0,t_f] \to \mathbb{R}^n$ is the state trajectory, $\dot{x} : [0,t_f] \to \mathbb{R}^n$ its derivative w. r. t. the time parameter $t \in [0,t_f]$, and $u : [0,t_f] \to \mathbb{R}^m$ the control trajectory. J is a vector of objective functionals,

$$J(x,u) = (J_1(x,u),\dots,J_k(x,u))^T$$

with $J_i(x,u) = \int_0^{t_f} C_i(x(t),u(t))\,dt$, $i = 1,\dots,k$. In contrast to the problems we considered before, the objective function depends on functions $x(t)$ and $u(t)$ rather than on single parameters. Our principal approach to solve such a trajectory optimization problem is to transform it into a nonlinear multiobjective optimization problem and solve this problem numerically using the image-set oriented recover algorithm. Such a transformation typically bases on a discretization in time such that the time-dependent functions are represented by a sequence of discrete state and control parameters that are approximations to the trajectories. (For an overview of different discretization techniques for single objective optimal control methods we refer e. g. to [2].) Thus, the mulitiobjective optimal control problem is transformed into a multiobjective optimization problem with many parameters, which consist of all time-discrete states and controls. To handle such high-dimensional multiobjective optimization problems, the presented image-set oriented recover algorithm is appropriate since it works in the low-dimensional image space rather than in the high-dimensional parameter space.

There are different possibilities how to transform the multiobjective optimal control problem into the multiobjective optimization problem, which highly depend on the system under consideration. In the following we will focus on differentially flat systems on the one hand and Lagrangian systems in general on the other hand. We will describe how the multiobjective optimal control problem can be transformed and show how these procedures can be applied to special mechatronical and mechanical systems.

5.5.1 Differentially Flat Systems

Differentially flat systems have the property that the inputs and states can be represented as a function of the flat outputs and a finite number of their derivatives wrt time (cf. [19]):

Definition 5.3 (Differential flatness [19, 43]). A system

$$\dot{x}(t) = g(x(t), u(t))$$

with states $x(t) \in \mathbb{R}^n$ and controls $u(t) \in \mathbb{R}^m$ is called *differentially flat* if there exists a fictitious output $y(t) \in \mathbb{R}^m$ with

$$\begin{aligned} &y = h(x, u, \dot{u}, \ddot{u}, \ldots, u^{(p)}) \text{ such that} \\ &x = \alpha(y, \dot{y}, \ddot{y}, \ldots, y^{(q)}) \text{ and } u = \beta(y, \dot{y}, \ddot{y}, \ldots, y^{(q)}). \end{aligned} \tag{5.40}$$

Here, h, α and β denote real-analytic functions and $p, q \in \mathbb{N}$. y is called a *flat output*.

Differentially flat system can especially be utilized in trajectory optimization (cf. [61]). The big advantage is that in this case the trajectories can be optimized in the space of the outputs y and afterwards, the corresponding inputs and states can be computed.

Thus, a single objective optimal control problem of the form

$$\begin{aligned} &\min_{x,u} j(x, u) \\ &\text{s. t. } \dot{x}(t) = g(x(t), u(t)) \end{aligned} \tag{5.41}$$

with a differentially flat system $\dot{x}(t) = g(x(t), u(t))$ with $x(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}^m$ can be transformed into an optimization problem of the form

$$\min_{y} f(y), \tag{5.42}$$

where $y(t) \in \mathbb{R}^m$ denotes the flat output (cf. e. g. [61, 50] and [41]).

This concept can be easily extended to the case of a vector-valued objective functional. In this case, the optimal control problem is transformed into a conventional multiobjective optimization problem of the form

$$\min_{y} F(y), \tag{5.43}$$

where F maps from $\mathbb{R}^m$ to $\mathbb{R}^k$ and $y(t) \in \mathbb{R}^m$ denotes the flat output.

Example 5.9 (Multiobjective optimization of the guidance of a rail-bound vehicle). In the following we consider the trajectory optimization of the *guidance module* of a rail-bound vehicle (cf. [21, 22]). More precisely, this guidance module is contained in the *RailCab* vehicle, which is a linear-motor driven railway system developed by the project RailCab ("Neue Bahntechnik Paderborn", [44]) at the University of Paderborn, Germany. Figure 5.12 displays the test vehicle. It belongs to a test facility with a track length of about 530 m. The track includes a novel passive switch which allows the processing of closely following vehicles. The vehicle itself consists of a superstructure that carries the load and two undercarriages. Among many other modules the RailCab is equipped with a guidance module which is based on one

Fig. 5.12 Photograph of the RailCab test vehicle

single wheel set. It enables a driving with low attrition and allows to use the novel concept for a passive switch (cf. [25]). The guidance module allows to actively control the lateral displacement of the RailCab vehicle in the rails. Within a given clearance, the RailCab can be moved freely. This is very important, because track laying does not result in ideally straight rails and flange strikes, i. e. bumpings of the rail-heads against the flanges, have to be avoided because they cause noise and wear on the wheels and rails. Figure 5.13 shows a typical rail and a sketch of the clearance, which is the maximum distance between the flanges and the rail-heads. We assume that the measured position of the rails (the track centerline) is known a priori.

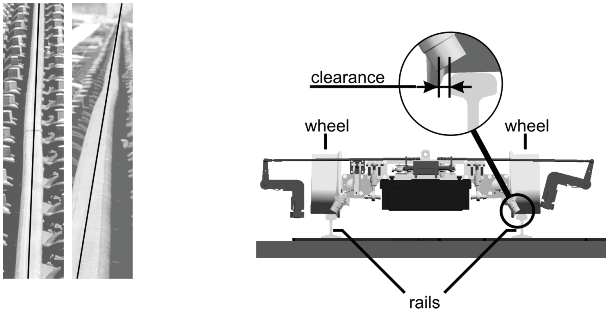


Fig. 5.13 Photograph of a rail (on the left) and sketch of the clearance (on the right) [21]

Within the clearance, the RailCab can be steered along arbitrary reference trajectories. The challenge was to compute Pareto optimal trajectories that meet several aims:

1. minimize the deviation of the vehicle from the track centerline, i. e. maximize "safety",
2. maximize the passenger comfort,
3. minimize the average energy consumption of the hydraulic actuators.

Based on a linear model of fourth order for the lateral dynamics of the RailCab vehicle (see [21] for more details), a multiobjective optimal control problem is formulated. In this model, the controlled outputs are flat. The desired reference trajectories of length s_h for both the front and the rear axle are approximated by cubic splines. For the computation of Pareto optimal trajectories, the image set-oriented recover algorithm has been used. The fact that the RailCab has to stay within the clearance is included as a constraint. Figure 5.14 shows an approximation of the Pareto front for an exemplary track section with a length of 8 m computed by the image-set oriented recover algorithm. To each point within this Pareto front corresponds a Pareto optimal trajectory on which the RailCab vehicle can be steered.

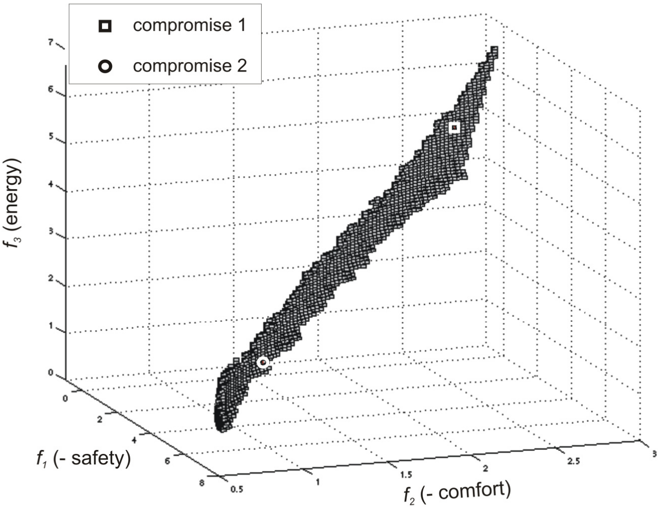


Fig. 5.14 Approximation of the Pareto front for the guidance module [21]

Two points within the Pareto front have been chosen (marked by a circle and a square) to demonstrate the results. The circle is an example for a more safe trajectory and the square for a more comfortable one. In Figure 5.15 the corresponding trajectories and the trajectories which stem from single objective optimizations for each of the three objectives are given. (Here, only the optimized interpolation values at the knot points, connected with lines, are plotted.) The black line is the track centerline and the gray lines describe the clearance around it.

One can observe that the trajectory which is more safe (line with circles) stays close to the centerline whereas the more comfortable trajectory (line with squares)

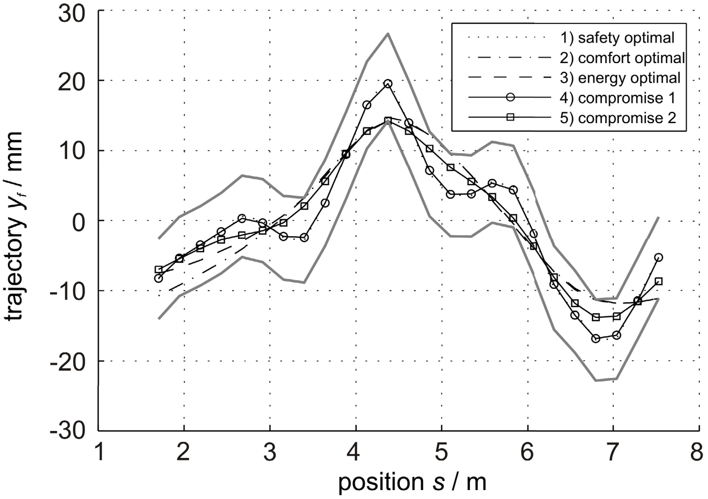


Fig. 5.15 Examples of Pareto optimal trajectories for the RailCab vehicle [21]

"cuts the corners" and is smoother. As expected, the energy optimal (dashed) and comfort optimal (dash-dot) trajectories lie close together.

5.5.2 Lagrangian Systems

We consider special kinds of dynamical systems $\dot{x}(t) = g(x(t), u(t))$, namely those systems that can be derived from a variational principle. In particular, we are interested in Lagrangian systems which comprise e. g. mechanical, but also electrical or mechatronical systems. In order to solve optimal control problems for those systems, we use DMOC (Discrete Mechanics and Optimal Control [48]), a technique that relies on a direct discretization of the variational formulation of the dynamics of the system. Based on the discretization the problem is transformed into a finite dimensional constrained optimization problem. The principal approach can be extended to the case of optimal control problems with multiple objectives. For convenience, we briefly summarize the basic idea.

Let M be an n-dimensional configuration manifold with tangent bundle TM and cotangent bundle T^*M. Consider a mechanical system with time-dependent configuration vector $q(t) \in M$ and velocity vector $\dot{q}(t) \in T_{q(t)}M$, $t \in [0, t_f]$, whose dynamical behavior is described by the Lagrangian $L : TM \to \mathbb{R}$. Typically, the Lagrangian L consists of the difference of the kinetic and potential energy. In addition, a force $f : TM \times U \to T^*M$ depending on a time-dependent control parameter $u(t) \in U \subseteq \mathbb{R}^m$ influences the system's motion. The aim is to move the mechanical system on a curve $q(t) \in M$, $t \in [0, t_f]$, from an initial state $(q^0, \dot{q}^0)$ to a final state $(q^{t_f}, \dot{q}^{t_f})$ under the influence of $f(q, \dot{q}, u)$ such that the curves q and u minimize a given objective functional

$$J(q, \dot{q}, u) = \int_0^{t_f} C(q(t), \dot{q}(t), u(t))\, dt \tag{5.44}$$

with $C : TM \times U \to \mathbb{R}^k$. Note, that the objective functional J involves k single objective functionals given as $J(q, \dot{q}, u) = (J_1(q, \dot{q}, u), \ldots, J_k(q, \dot{q}, u))^T$ according to (5.39) with

$$J_i(q, \dot{q}, u) = \int_0^{t_f} C_i(q(t), \dot{q}(t), u(t))\, dt, \quad i = 1, \ldots, k,$$

and $C(q, \dot{q}, u) = (C_1(q, \dot{q}, u), \ldots, C_k(q, \dot{q}, u))^T$. At the same time, the motion $q(t)$ has to satisfy the *Lagrange-d'Alembert principle*, which requires that

$$\delta \int_0^{t_f} L(q(t), \dot{q}(t))\, dt + \int_0^{t_f} f(q(t), \dot{q}(t), u(t)))\, \delta q\, dt = 0 \tag{5.45}$$

for all variations δq with $\delta q(0) = \delta q(t_f) = 0$. The principle (5.45) is equivalent to the forced Euler-Lagrange equations

$$\frac{d}{dt}\frac{\partial}{\partial \dot{q}} L(q, \dot{q}) - \frac{\partial}{\partial q} L(q, \dot{q}) = f(q, \dot{q}, u), \tag{5.46}$$

which provide as system of differential equations the equations of motion that can be summarized in the general form $\dot{x} = g(x, u)$ with $x = (q, \dot{q})$.

The optimal control problem consisting of minimizing (5.44) subject to (5.46) is numerically solved using a direct discretization approach [39, 48]. The state space TM is replaced by $M \times M$ and a path $q : [0, t_f] \to M$ by a *discrete path* $q_d : \{0, h, 2h, \ldots, Nh = t_f\} \to M$, with time step h and N a positive integer such that $q_k = q_d(kh)$ is an approximation to $q(kh)$. Similar, the control function $u : [0, t_f] \to U$ is replaced by a discrete control function $u_d : \{0, h, 2h, \ldots, Nh = t_f\} \to U$, approximating the control on each interval $[kh, (k+1)h]$ by a discrete control u_k (writing $u_k = u_d((k + \frac{1}{2})h)$).

Via an approximation of the action integral in (5.45) by a *discrete Lagrangian* $L_d : M \times M \to \mathbb{R}$,

$$L_d(q_k, q_{k+1}) \approx \int_{kh}^{(k+1)h} L(q(t), \dot{q}(t))\, dt, \tag{5.47}$$

and *discrete forces*

$$f_k^- \cdot \delta q_k + f_k^+ \cdot \delta q_{k-1} \approx \int_{kh}^{(k+1)h} f(q(t), \dot{q}(t), u(t)) \cdot \delta q(t)\, dt, \tag{5.48}$$

where the left and discrete forces $f_k^\pm$ now depend on (q_k, q_{k+1}, u_k) we obtain the *discrete Lagrange-d'Alembert principle* (5.49). This requires to find discrete paths $\{q_k\}_{k=0}^N$ such that for all variations $\{\delta q_k\}_{k=0}^N$ with $\delta q_0 = \delta q_N = 0$, one has

$$\delta \sum_{k=0}^{N-1} L_d(q_k, q_{k+1}) + \sum_{k=0}^{N-1} f_k^- \cdot \delta q_k + f_k^+ \cdot \delta q_{k+1} = 0, \tag{5.49}$$

which is equivalent to the *forced discrete Euler-Lagrange equations*

$$D_2 L_d(q_{k-1}, q_k) + D_1 L_d(q_k, q_{k+1}) + f_{k-1}^+ + f_k^- = 0, \quad k = 1, \ldots, N-1, \tag{5.50}$$

where D_i denotes the derivative w. r. t. the i-th argument. In the same manner we obtain via an approximation of the objective functional (5.44) the *discrete objective function* $J_d(q_d, u_d)$, such that we can formulate the *Discrete Constrained Multiobjective Optimization Problem* as

$$\min_{q_d, u_d} J_d(q_d, u_d) = \sum_{k=0}^{N-1} C_d(q_k, q_{k+1}, u_k) \tag{5.51}$$

subject to the discretized boundary constraints and the forced discrete Euler-Lagrange equations (5.50). Here, it holds again $J_d = (J_{d,1}, \ldots, J_{d,k})^T$ and $C_d = (C_{d,1}, \ldots, C_{d,k})^T$, where $J_{d,i}$ and $C_{d,i}$ are approximations to J_i and C_i, respectively, with $i = 1, \ldots, k$. The number of the optimization parameters $q_d = (q_0, \ldots, q_N)$ and $u_d = (u_0, \ldots, u_{N-1})$ as well as the number of the equality constraints (the forced discrete Euler-Lagrange equations) of this nonlinear multiobjective optimization problem depend on the discrete grid that is used for the approximation. To meet accuracy requirements of the approximated trajectories (for a detailed convergence analysis dependent on the quadrature rules used in (5.47) and (5.48) we refer to [48]), typically a fine grid which corresponds to a small time step h is chosen, which leads to a high number of optimization parameters and equality constraints, whereas the number of objective functions J_d is independent of the time step. Thus, the image-set oriented methods described before are suitable to numerically solve this high-dimensional multiobjective optimization problem.

Example 5.10 (Underwater glider). As an application for multiobjective optimal control we consider a class of Autonomous Underwater Vehicles (AUVs) known as gliders. In order to keep the gliders autonomously operational for the greatest amount of time, it is important to minimize the amount of energy the gliders use for transport and - at the same time - minimize the time of operation when specific maneuvers are performed. The problem considered here is to find an optimal trajectory of a glider that needs to move from one location to another within a prescribed current (cf. [47]). The glider is assumed to be actuated by a gyroscopic force which implies that the relative forward speed of the glider is constant. However, the orientation of the glider cannot change instantly and the control force induces the change in the orientation of the glider. In addition to the minimization of the amount of control effort, the goal is to identify trajectories that are also time-optimal, such that the glider needs as little time as possible to reach the final destination. Thus, we have to consider a multiobjective optimization problem with the two objectives *minimize control effort* and *minimize duration time of the maneuver*. As in [63] the glider is modeled as a pointmass (with normalized mass equal to 1) in $\mathbb{R}^2$ and actuated by a gyroscopic force acting orthogonal to the relative velocity between fluid and body. Let $q(t) = (x(t), y(t))$ be the glider position, $\dot{q}(t) = (\dot{x}(t), \dot{y}(t))$ the absolute glider velocity, $V(t) = (V_x(t), V_y(t))$ the current velocity field, and $u(t) \in \mathbb{R}$ the control input representing the change in the orientation. By introducing $\dot{q}_{rel}(t) = (\dot{q}(t) - V(t))$ as relative velocity, the Lagrangian $L(q_{rel}(t), \dot{q}_{rel}(t)) = \frac{1}{2}\|\dot{q}_{rel}(t)\|^2$ in the body fixed frame is the kinetic energy of the relative motion of the glider. The gyroscopic force acting on the system is given by $f(q_{rel}(t), \dot{q}_{rel}(t), u(t)) = \left(-u(t)\,\dot{q}_{rel,y}(t), u(t)\,\dot{q}_{rel,x}(t)\right)^T$. The resulting Euler-Lagrange equations read as

$$\begin{aligned}\ddot{x}(t) &= -u(t)(\dot{y}(t) - V_y(t)) + \dot{V}_x(t),\\ \ddot{y}(t) &= u(t)(\dot{x}(t) - V_x(t)) + \dot{V}_y(t).\end{aligned}$$

The glider has to be steered within the time span $[0, t_f]$ with free final time t_f from an initial configuration $q(0) = q^0$ to a final one $q(t_f) = q^{t_f}$, optimally with respect to the vector-valued objective functional

$$J = \begin{pmatrix} \int_0^{t_f} \|f(q_{rel}(t), \dot{q}_{rel}(t), u(t))\|^2\, dt \\ \int_0^{t_f} 1\, dt \end{pmatrix}.$$

In the discrete setting we model the free final time by a variable step size h that acts as an additional optimization variable bounded as $0 < h \leq h_{max}$ to ensure positive step size and solutions of desired accuracy. For a fixed number of discretization points the final time is then given by $t_f = (N-1)h$. As initial constraint we assume a prescribed initial configuration $q_{rel}(0) = (10, 0)$ and an initial relative velocity as $\dot{q}_{rel}(0) = (-10, -10)$. The final configuration is given by $q_{rel}(t_f) = (15, 2)$, while the final relative velocity is free with same magnitude as the initial one, as the control force only influences the orientation, rather than the magnitude of the relative velocity. The current velocity is assumed to be configuration-dependent in x- and zero in y-direction given as $V = (x, 0)$. The discretization of the glider model leads to a constrained nonlinear multiobjective optimization problem. It is solved making use of the image-set oriented recover algorithm. Here, the distance of the objective values to the targets is optimized subject to the constraints stemming from the system dynamics. Figure 5.16 shows the results: the approximated Pareto front (left) and some corresponding trajectories (right). As expected, the control effort increases for decreasing maneuver time. Comparing different trajectories corresponding to different Pareto points, all trajectories show the same qualitative behavior: As the initial velocity is directed away from the destination, the gyroscopic control force enforces the glider performing a circular motion starting in direction of the initial velocity. Due to the fluid velocity in x-direction, the glider moves along a loop to reach the desired final location as depicted on the right in Figure 5.16. For trajectories with shorter time duration the loop becomes smaller and the corresponding control effort becomes higher since a big change of orientation in short time requires a high force applied to the system.

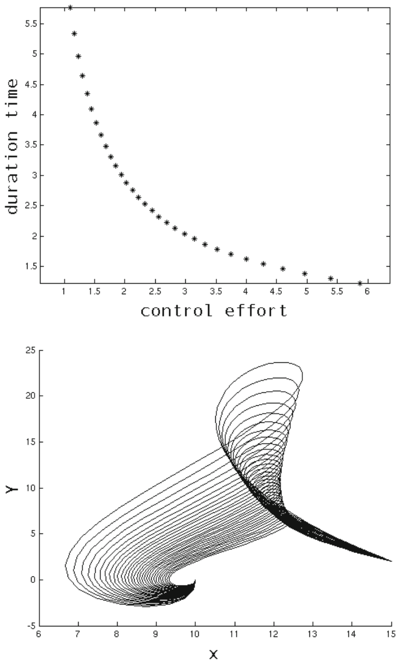


Fig. 5.16 Pareto front for the underwater glider computed with the image-set oriented recover algorithm (left), and the corresponding trajectories in configuration space (right)

5.6 Concluding Remarks

In this chapter, we have given an overview of recently developed set oriented methods for the numerical treatment of MOPs. The characteristic of these methods is that they generate box collections that aim for tight coverings of the Pareto set (or front) of a given MOP. The methods are divided into subdivision techniques and a particular kind of continuation methods (recover techniques). Subdivision techniques generate a sequence of nested box collections that converges (ideally) to the Pareto set, and the idea of the recover techniques is to extend a given collection $\mathscr{C}$ by a local search which is performed around promising elements (boxes) of $\mathscr{C}$. Subdivision techniques are of global nature and highly competitive to other state-of-the-art methods in particular if the dimension of the parameter space is moderate (say, $n < 50$), and the number of objectives is low ($k < 5$). Continuation methods are of local nature, but in turn applicable to higher dimensional problems ($n \gg 1000$). The latter has been demonstrated on several multiobjective optimal control problems which were transformed into high-dimensional MOPs.

Acknowledgements. The first author acknowledges support from CONACyT project no. 128554. This work was partly developed in the course of the Collaborative Research Center 614 – Self-Optimizing Concepts and Structures in Mechanical Engineering – University of Paderborn, and was partly funded by the Deutsche Forschungsgemeinschaft. The authors would like to thank everyone who supported them in the design and analysis of the above mentioned algorithms. Very special thanks go to Alesandro Dell'Aere, Stefan Sertl, Maik Ringkamp and Albert Seifried. Additionally we would like to thank the Chair of Control Engineering and Mechatronics (Prof. Dr.-Ing. A. Trächtler) and the Chair of Power Electronics and Electrical Drives (Prof. Dr.-Ing. J. Böcker), University of Paderborn, Germany who provided the technical applications presented in this work. In particular, we thank Jens Geisler and Tobias Knoke.

References

1. Allgower, E.L., Georg, K.: Numerical Continuation Methods. Springer (1990)
2. Betts, J.T.: Survey of numerical methods for trajectory optimization. AIAA J. Guidance, Control, and Dynamics 21(2), 193–207 (1998)
3. Binder, T., Blank, L., Bock, H.G., Bulirsch, R., Dahmen, W., Diehl, M., Kronseder, T., Marquardt, W., Schlöder, J.P., von Stryk, O.: Introduction to model based optimization of chemical processes on moving horizons. In: Grötschel, M., Krumke, S.O., Rambau, J. (eds.) Online Optimization of Large Scale Systems: State of the Art, pp. 295–340. Springer (2001)
4. Blesken, M., Rückert, U., Steenken, D., Witting, K., Dellnitz, M.: Multiobjective Optimization for Transistor Sizing of CMOS Logic Standard Cells Using Set-Oriented Numerical Techniques. In: 27th Norchip Conference (2009)
5. Bosman, P.A.N., de Jong, E.D.: Exploiting gradient information in numerical multi-objective evolutionary optimization. In: Genetic and Evolutionary Computation Conference - GECCO 2005. ACM (2005)
6. Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd edn. Springer, New York (2007) ISBN 978-0-387-33254-3
7. Das, I., Dennis, J.: Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. SIAM Journal of Optimization 8, 631–657 (1998)
8. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, Chichester (2001) ISBN 0-471-87339-X
9. Dell'Aere, A.: Multi-objective optimization in self-optimizing systems. In: Proceedings of the IEEE 32nd Annual Conference on Industrial Electronics (IECON), pp. 4755–4760 (2006)
10. Dell'Aere, A.: Numerical methods for the solution of bi-level multi-objective optimization problems. PhD thesis. University of Paderborn, Germany (2008)
11. Dellnitz, M., Hohmann, A.: The computation of unstable manifolds using subdivision and continuation. In: Broer, H.W., van Gils, S.A., Hoveijn, I., Takens, F. (eds.) Nonlinear Dynamical Systems and Chaos, vol. 19, pp. 449–459. PNLDE, Birkhäuser (1996)
12. Dellnitz, M., Hohmann, A.: A subdivision algorithm for the computation of unstable manifolds and global attractors. Numerische Mathematik 75, 293–317 (1997)

13. Dellnitz, M., Ober-Blöbaum, S., Post, M., Schütze, O., Thiere, B.: A multi-objective approach to the design of low thrust space trajectories using optimal control. Celestial Mechanics and Dynamical Astronomy 105(1), 33–59 (2009)
14. Dellnitz, M., Schütze, O., Hestermeyer, T.: Covering Pareto sets by multilevel subdivision techniques. Journal of Optimization Theory and Applications 124, 113–155 (2005)
15. Dellnitz, M., Schütze, O., Sertl, S.: Finding zeros by multilevel subdivision techniques. IMA Journal of Numerical Analysis 22(2), 167–185 (2002)
16. Eichfelder, G.: Adaptive Scalarization Methods in Multiobjective Optimization. Springer, Heidelberg (2008) ISBN 978-3-540-79157-7
17. Fliege, J.: Gap-free computation of Pareto-points by quadratic scalarizations. Mathematical Methods of Operations Research 59, 69–89 (2004)
18. Fliege, J., Fux Svaiter, B.: Steepest descent methods for multicriteria optimization. Mathematical Methods of Operations Research 51(3), 479–494 (2000)
19. Fliess, M., Levine, J., Martin, P., Rouchon, P.: Flatness and defect of non-linear systems: Introductory theory and examples. International Journal of Control 61(6), 1327–1361 (1995)
20. Gandibleux, X.: Metaheuristics for Multiobjective Optimisation. Lecture notes in economics and mathematical systems. Springer (2004)
21. Geisler, J., Witting, K., Trächtler, A., Dellnitz, M.: Multiobjective optimization of control trajectories for the guidance of a rail-bound vehicle. In: 17th IFAC World Congress, Seoul, Korea, July 6-11 (2008)
22. Geisler, J., Witting, K., Trächtler, A., Dellnitz, M.: Self-optimization of the guidance module of a rail-bound vehicle. In: Gausemeier, J., Rammig, F., Schäfer, W. (eds.) 7th International Heinz Nixdorf Symposium 'Self-optimizing Mechatronic Systems: Design the Future', February 20-21, pp. 85–100. HNI-Verlagsschriftenreihe, Paderborn (2008)
23. Harada, K., Sakuma, J., Kobayashi, S., Ono, I.: Uniform sampling of local Pareto-optimal solution curves by pareto path following and its applications in multi-objective GA. In: GECCO, pp. 813–820 (2007)
24. Henderson, M.E.: Multiple parameter continuation: Computing implicitly defined k-manifolds. International Journal of Bifurcation and Chaos 12, 451–476 (2002)
25. Hestermeyer, T., Schlautmann, P., Ettingshausen, C.: Active suspension system for railway vehicles – system design and kinematics. In: 2nd IFAC - Conference on Mechatronic Systems, Berkeley, California, USA (2002)
26. Hillermeier, C.: Nonlinear Multiobjective Optimization - A Generalized Homotopy Approach. Birkhäuser (2001)
27. Horst, R., Tuy, H.: Global Optimization: Deterministic Approaches. Springer (1993)
28. Hsu, C.S.: Cell-to-cell mapping: a method of global analysis for nonlinear systems. Applied mathematical Sciences. Springer (1987)
29. Hsu, H.C.: Global analysis by cell mapping. International Journal of Bifurcation and Chaos 2, 727–771 (1992)
30. Junge, O., Marsden, J.E., Ober-Blöbaum, S.: Optimal Reconfiguration of Formation Flying Spacecraft - a decentralized approach. In: IEEE Conference on Decision and Control, San Diego, CA, USA, pp. 5210–5215 (2006)
31. Karush, W.E.: Minima of functions of several variables with inequalities as side conditions. PhD thesis, University of Chicago (1939)
32. Klamroth, K., Tind, J., Wiecek, M.: Unbiased approximation in multicriteria optimization. Mathematical Methods of Operations Research 56, 413–437 (2002)

33. Knoke, T., Romaus, C., Böcker, J., Dell'Aere, A., Witting, K.: Energy management for an onboard storage system based on multiobjective optimization. In: Proceedings of the IEEE 32nd Annual Conference on Industrial Electronics (IECON), pp. 4677–4682 (2006)
34. Krüger, M., Witting, K., Trächtler, A., Dellnitz, M.: Parametric model order reduction in hierarchical multiobjective optimization of mechatronic systems. In: 18th IFAC World Congress, Milano, Italy, August 28-September 2 (2010)
35. Kuhn, H., Tucker, A.: Nonlinear programming. In: Neumann, J. (ed.) Proceeding of the 2nd Berkeley Symposium on Mathematical Statistics and Probability (1951)
36. Leyendecker, S., Ober-Blöbaum, S., Marsden, J.E., Ortiz, M.: Discrete mechanics and optimal control for constrained systems. Optimal Control, Applications and Methods 31(6), 505–528 (2010)
37. Li, R., Pottharst, A., Witting, K., Znamenshchykov, O., Böcker, J., Fröhleke, N., Feldmann, R., Dellnitz, M.: Design and implementation of a hybrid energy supply system for railway vehicles. In: Proc. of APEC2005, IEEE Applied Power Electronics Conference 2005, Austin, Texas, USA, pp. 474–480 (2005)
38. Logist, F., Houska, B., Diehl, M., Van Impe, J.: Fast pareto set generation for nonlinear optimal control problems with multiple objectives. Structural and Multidisciplinary Optimization 42, 591–603 (2010), doi:10.1007/s00158-010-0506-x
39. Marsden, J.E., West, M.: Discrete mechanics and variational integrators. Acta Numerica 10, 357–514 (2001)
40. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer Academic Publishers, Boston (1999)
41. Milam, M., Mushambi, K., Murray, R.: A new computational approach to real-time trajectory generation for constrained mechanical systems. In: Proceedings of the 39th IEEE Conference on Decision and Control, vol. 1, pp. 845–551 (2000)
42. Moore, A., Ober-Blöbaum, S., Marsden, J.E.: Optimization of spacecraft trajectories: a method combining invariant manifold techniques and discrete mechanics and optimal. In: 19th AAS/AIAA Space Flight Mechanics Meeting, Februar 8-12 (2009)
43. Murray, R., Rathinam, M., Sluis, W.: Differential flatness of mechanical control systems: A catalog of prototype systems. In: Proceedings of the 1995 ASME International Congress and Exposition, San Francisco (1995)
44. Neue Bahntechnik Paderborn – Project. Web-Page, `http://www.railcab.de`
45. Nocedal, J., Wright, S.: Numerical Optimization. Springer Series in Operations Research and Financial Engineering. Springer (2006)
46. Numerical Algorithms Group (C-Library). Web-Page, `http://www.nag.co.uk/numeric/CL/CLdescription.asp`
47. Ober-Blöbaum, S.: Discrete mechanics and optimal control. PhD thesis. University of Paderborn (2008)
48. Ober-Blöbaum, S., Junge, O., Marsden, J.E.: Discrete mechanics and optimal control: an analysis. Control, Optimisation and Calculus of Variations 17(2), 322–352 (2011)
49. Ober-Blöbaum, S., Timmermann, J.: Optimal control for a pitcher's motion modeled as constrained mechanical system. In: 7th International Confenrence on Multibody Systems, Nonlinear Dynamics, and Control, ASME International Design Engineering Technical Conferences, San Diego, CA, USA (2009)
50. Petit, N., Milam, M., Murray, R.: Inversion based constrained trajectory optimization. In: 5th IFAC Symposium on Nonlinear Control Systems (2001)

51. Pottharst, A., Baptist, K., Schütze, O., Böcker, J., Fröhlecke, N., Dellnitz, M.: Operating point assignment of a linear motor driven vehicle using multiobjective optimization methods. In: Proceedings of the 11th International Conference EPE-PEMC 2004, Riga, Latvia (2004)
52. Recchioni, M.C.: A path following method for box-constrained multiobjective optimization with applications to goal programming problems. Mathematical Methods of Operations Research 58, 69–85 (2003)
53. Rheinboldt, W.: On the computation of multi-dimensional solution manifolds of parametrized equations. Numerische Mathematik 53, 165–181 (1988)
54. 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. To appear in Engineering Optimization (2011)
55. Schäffler, S., Schultz, R., Weinzierl, K.: A stochastic method for the solution of unconstrained vector optimization problems. Journal of Optimization Theory and Applications 114(1), 209–222 (2002)
56. Schneider, T., Schulz, B., Henke, C., Witting, K., Steenken, D., Böcker, J.: Energy transfer via linear doubly-fed motor in different operating modes. In: Proceedings of the International Electric Machines and Drives Conference, Miami, Florida, USA, May 3-6 (2009)
57. Schütze, O.: Set Oriented Methods for Global Optimization. PhD thesis, University of Paderborn (2004),
http://ubdata.uni-paderborn.de/ediss/17/2004/schuetze/
58. Schütze, O., Dell'Aere, A., Dellnitz, M.: On continuation methods for the numerical treatment of multi-objective optimization problems. In: Branke, J., Deb, K., Miettinen, K., Steuer, R.E. (eds.) Practical Approaches to Multi-Objective Optimization. Dagstuhl Seminar Proceedings, vol. 04461. Internationales Begegnungs- und Forschungszentrum (IBFI), Schloss Dagstuhl (2005),
http://drops.dagstuhl.de/opus/volltexte/2005/349
59. Schütze, O., Jourdan, L., Legrand, T., Talbi, E.-G., Wojkiewicz, J.L.: New analysis of the optimization of electromagnetic shielding properties using conducting polymers and a multi-objective approach. Polymers for Advanced Technologies 19, 762–769 (2008)
60. Schütze, O., Vasile, M., Junge, O., Dellnitz, M., Izzo, D.: Designing optimal low thrust gravity assist trajectories using space pruning and a multi-objective approach. Engineering Optimization 41(2), 155–181 (2009)
61. Van Nieuwstadt, M.J., Murray, R.M.: Real-time trajectory generation for differentially flat systems. International Journal of Robust and Nonlinear Control 8(11), 995–1020 (1998)
62. Witting, K., Schulz, B., Dellnitz, M., Böcker, J., Fröhleke, N.: A new approach for online multiobjective optimization of mechatronic systems. International Journal on Software Tools for Technology Transfer STTT 10(3), 223–231 (2008)
63. Zhang, W., Inanc, T., Ober-Blöbaum, S., Marsden, J.E.: Optimal trajectory generation for a dynamic glider in ocean flows modeled by 3D B-Spline functions. In: IEEE International Conference on Robotics and Automation (ICRA), Pasadena, CA, USA (2008)