JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 124, No. 1, pp. 113–136, January 2005 (© 2005)

Covering Pareto Sets by Multilevel Subdivision Techniques[1]

M. Dellnitz,[2] O. Schütze,[3] and T. Hestermeyer[4]

Communicated by R. Glowinski

Abstract. In this work, we present a new set-oriented numerical method for the numerical solution of multiobjective optimization problems. These methods are global in nature and allow to approximate the entire set of (global) Pareto points. After proving convergence of an associated abstract subdivision procedure, we use this result as a basis for the development of three different algorithms. We consider also appropriate combinations of them in order to improve the total performance. Finally, we illustrate the efficiency of these techniques via academic examples plus a real technical application, namely, the optimization of an active suspension system for cars.

Key Words. Multiobjective optimization, Pareto sets, global optimization.

1. Introduction

In a variety of applications in industry or economy, the problem arises that several objective functions have to be optimized at the same time. For instance, for a perfect economical production plan, one wants to simultaneously minimize the cost and maximize the quality. This example

[1]The authors thank Joachim Lückel for his suggestion to get into the interesting field of multiobjective optimization. Katrin Baptist as well as Frank Scharfeld helped the authors with fruitful discussions. This work was partly supported by the Deutsche Forschungsgemeinschaft within SFB 376 and SFB 614.

[2]Professor, Institute for Mathematics, University of Paderborn, Paderborn, Germany.

[3]Graduate Student, Institute for Mathematics, University of Paderborn, Paderborn, Germany.

[4]Graduate Student, Mechatronics Laboratory Paderborn, University of Paderborn, Paderborn, Germany.

illustrates a natural feature of these problems, namely, that typically the different objectives contradict each other and therefore do not have identical optima. Thus, the question arises of how to approximate the optimal compromises and this leads to the problem of multiobjective optimization (MOP).

In an (MOP), there are given k objective functions $f_1, \ldots, f_k : \mathbb{R}^n \to \mathbb{R}$ which have to be minimized. The set of optimal compromises with respect to the objective functions is called the Pareto set. Mathematically speaking, a point $x \in \mathbb{R}^n$ in parameter space is a Pareto point if there is no other point which is at least as good as x in all the objectives and is strictly better in at least one objective. Thus, the numerical solution of an (MOP) consists in finding an approximation to the Pareto set.

Multiobjective optimization is currently a very active area of research; see e.g. Refs. 1–5 and references therein. Accordingly, there exists a variety of different approaches for solving these problems. An excellent overview of deterministic techniques is given in Ref. 1; these methods are similar in spirit to the traditional optimization techniques for classical (scalar) optimization problems. Another prominent approach is based on evolutionary algorithms (Refs. 2–5); these methods are particularly advantageous in the situation where the (MOP) is discrete.

A method which is based on a stochastic approach is presented in Ref. 7. More concretely, there the authors derive a stochastic differential equation (SDE) for which typical solutions stay close to the Pareto set for a relatively long period of time. This knowledge can then be used to locate the Pareto set via a solution of the (SDE). Similar to the evolutionary strategies, here the idea is to approximate directly the entire Pareto set and not just single Pareto points on the set.

In this article, we use also a global approach to compute approximations of the entire Pareto set. Our methods are set-oriented and are based on the concepts presented in e.g. Refs. 8–10. The underlying idea is to write down an iteration scheme which, interpreted as a discrete dynamical system, possesses the Pareto set as an attractor. Then, set-oriented numerical methods for dynamical systems can be used for its approximation. The dynamical systems (i.e., iteration schemes) that we are considering here are obtained via suitable discretizations of the differential equation derived in Ref. 7. However, in contrast to that work, we do not have to add randomness in our algorithms by a consideration of a related (SDE). Rather, we are working directly with the deterministic system itself.

More concretely, we propose three different set-oriented multilevel approaches for the approximation of the Pareto set. First, we apply the techniques from Ref. 10 more or less directly to the context of multiobjective optimization. That is, we present a subdivision algorithm for the

approximation of Pareto sets, which creates tight box coverings of these objects. We discuss also realizations and prove convergence results concerning the abstract subdivision process. Since it can occur that, in the course of the subdivision procedure, one loses part of the covering, we present a recovering algorithm which can be viewed as a postprocessing procedure for the subdivision scheme. This yields our second approach. In a third algorithm, we combine the creation of the box covering with appropriate branch and bound strategies which are similar in spirit to those of Ref. 11. Finally, we discuss a combination of all three algorithms such that the performance of the resulting algorithm is improved.

A detailed outline of the article is as follows. In Section 2, we recall the theoretical background for both multiobjective optimization problems and subdivision schemes in the context of global zero finding. Here, we define also the dynamical system which is the basis for our set-oriented numerical approach. In Section 3, we prove the convergence of certain iteration schemes toward the set of substationary points. A combination of this result with results from Ref. 10 yields the desired convergence result for our subdivision procedure. The corresponding algorithms, realizations and extensions are presented in Section 4. Finally, in Section 5, we illustrate the efficiency of our approach via several examples.

2. Theoretical Background

Now we summarize briefly the theoretical background which is necessary for understanding our considerations in the subsequent sections. There are two different mathematical topics that we have to address: the concept of Pareto optimality (Section 2.1) and a set-oriented numerical approach for a zero-finding procedure (Section 2.2).

2.1. Pareto Optimality.

Definition of Pareto Optimality. In a multiobjective optimization problem (MOP), the task is to optimize simultaneously k objective functions $f_1, \ldots, f_k : \mathbb{R}^n \to \mathbb{R}$. More precisely, an unconstrained (MOP) can be stated as follows:

$$\text{(MOP)} \quad \min_{x \in \mathbb{R}^n} \{F(x)\},$$

where the function F is defined as the vector of the objective functions

$$F : \mathbb{R}^n \to \mathbb{R}^k, \quad F(x) = (f_1(x), \ldots, f_k(x)).$$

Obviously, we have to define what is meant by finding the minimum of a vector-valued function in (MOP). For this, we state the following definition.

Definition 2.1. Let $v, w \in \mathbb{R}^k$. 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 in an analogous way.

Since $\leq_p$ defines just a partial order on $\mathbb{R}^n$, we cannot proceed as in the classical scalar case. In fact, we cannot expect to find isolated stationary points. Rather, we have to find the set of optimal compromises. Following Pareto (Ref. 12), these are defined in the following way.

Definition 2.2.

(a) Consider the multiobjective optimization problem of Section 2.1. A point $\bar{x} \in \mathbb{R}^n$ is called (globally) Pareto optimal or a (global) Pareto point if there is no $y \in \mathbb{R}^n$ such that

$$F(y) \neq F(\bar{x}) \quad \text{and} \quad F(y) \leq_p F(\bar{x}). \tag{1}$$

(b) A point $\bar{x} \in \mathbb{R}^n$ is a local Pareto point if there is a neighborhood $U(\bar{x}) \subset \mathbb{R}^n$ of $\bar{x}$ such that there is no $y \in U(\bar{x})$ satisfying (1).

Example 2.1. In Figure 1, we present an example of two objective functions $f_j : \mathbb{R} \to \mathbb{R}$, $j = 1, 2$. In this case, the set of local Pareto points consists of the union of the intervals [0, 1] and [1.5, 2]. However, only the points in the interval [1.5, 2] are also globally Pareto optimal.

Necessary Condition for Optimality. In our numerical methods, we are going to make use of the following theorem of Kuhn and Tucker (Ref. 13), which states a necessary condition for Pareto optimality.

Theorem 2.1. Let x^* be a Pareto point of (MOP). Then, there exist nonnegative scalars $\alpha_1, \ldots, \alpha_k \geq 0$ such that

$$\sum_{i=1}^{k} \alpha_i = 1 \quad \text{and} \quad \sum_{i=1}^{k} \alpha_i \nabla f_i(x^*) = 0. \tag{2}$$

Obviously, (2) is not a sufficient condition for (local) Pareto optimality. On the other hand, points satisfying (2) are certainly Pareto candidates; thus, following Ref. 1, we emphasize now their relevance via the following definition.

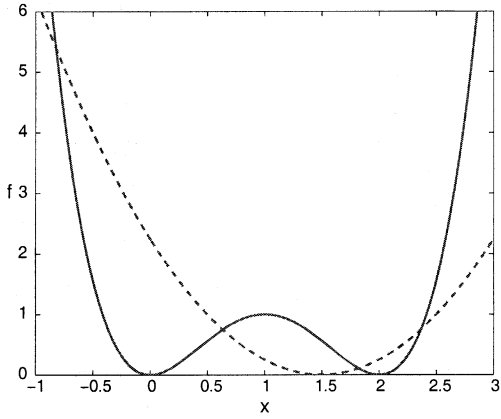

Fig. 1. Two objective functions $f_j : \mathbb{R} \to \mathbb{R}$ $(j = 1, 2)$ on the interval $[-1, 3]$.

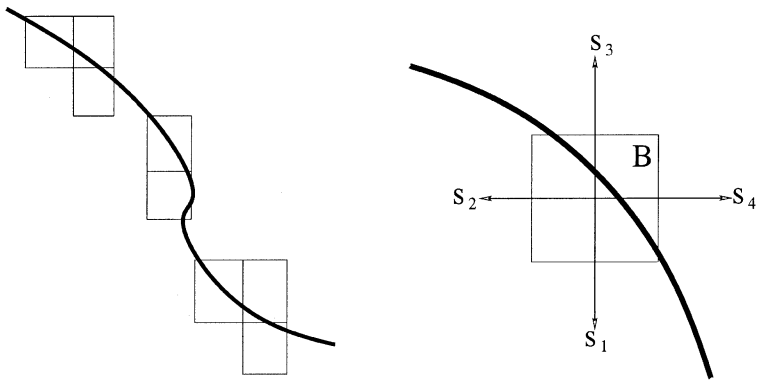

Fig. 2. Recovering algorithm: uncomplete covering of the Pareto set (left) and possible choice of test points for a given box B (right).

Definition 2.3. A point $x \in \mathbb{R}^n$ is called a substationary point if there exist scalars $\alpha_1, \ldots, \alpha_k \geq 0$ such that (2) is satisfied.

Remark 2.1. Under certain smoothness assumptions, it can be shown that the set of local substationary points defines typically a $(k-1)$-dimensional manifold, where k is the number of objectives described by F (see Ref. 6). Essentially, this is nothing else but an immediate consequence of the Implicit Function Theorem.

Descent Direction. Similar to classical iteration schemes for the numerical solution of scalar optimization problems, we need to identify a descent direction for our numerical methods. More precisely, for a point which is not substationary, we need to know a direction in $\mathbb{R}^n$ in which all the k objectives are simultaneously decreasing. For this purpose, we summarize briefly in the following the main results of Ref. 7. These results have to be viewed as the natural extension of the classical underlying theory for the method of steepest descent.

First, we associate with $F:\mathbb{R}^n \to \mathbb{R}^k$, $F(x)=(f_1(x),\ldots,f_k(x))$, the following quadratic optimization problem:

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

Then, we can show the following result.

Theorem 2.2. See Ref. 7. Let $q:\mathbb{R}^n \to \mathbb{R}^n$ be defined by

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

where $\hat{\alpha}$ is a solution of (QOP). Then, either $q(x)=0$ or $-q(x)$ is a descent direction for all objective functions $f_1,\ldots,f_k$ in x. Moreover, q is locally Lipschitz continuous.

By this result, the initial-value problem

$$\dot{x}(t)=-q(x(t)), \quad x(0)=x_0, \tag{3}$$

is well posed. Let $x:[0,\infty[\to\mathbb{R}^n$ be the unique solution of (3). Then, one can show that this solution satisfies

$$F(x(s)) \geq_p F(x(t)) \text{ and } F(x(s)) \neq F(x(t)), \quad \text{for all } 0\leq s<t<\infty.$$

In fact, suppose additionally that the set

$$R_{\leq_p}=\{x\in\mathbb{R}^n : F(x)\leq_p F(x_0)\}$$

is bounded. Then, the solution $x(t)$ has to converge to a substationary point for $t\to\infty$. Thus, a suitable discretization of the generalized gradient system (3) yields numerical iteration schemes converging toward substationary points; see Section 3.

Observe that, by Theorem 2.2, each Pareto point is a zero of the function q. Therefore, the aim is to find the set of zeros of q. Since these zeros are not isolated (cf. Remark 2.1), a set-oriented approach turns out to be most adequate for their approximation. Next, we describe the general approach and in particular the related subdivision procedure.

2.2. Subdivision Algorithm. Now we review briefly a subdivision algorithm for the approximation of the set of zeros of a function in a compact domain. For a detailed exposition, the reader is referred to Ref. 10. We consider a finite collection of discrete dynamical systems of the type

$$x_{j+1} = g_\ell(x_j), \quad j = 0, 1, 2, \ldots,$$

where we assume for simplicity that each $g_\ell : \mathbb{R}^n \to \mathbb{R}^n$, $\ell = 1, \ldots, s$, is a diffeomorphism.

Remark 2.2. In the applications that we have in mind, each function g_ℓ is an iteration scheme for finding the zeros of a given function. For instance, the functions g_ℓ can correspond to the damped Newton method with s different stepsizes. The subdivision algorithm presented now allows us to determine a close covering of the entire set of zeros.

The following definitions are taken from Ref. 10. Denote by

$$\Omega = \{1, 2, \ldots, s\}^{\mathbb{N}_0}$$

the set of all sequences of the symbols $\{1, 2, \ldots, s\}$. For $\omega = (\omega_i) \in \Omega$, set

$$\omega^j = (\omega_0, \omega_1, \ldots, \omega_{j-1})$$

and for $j \geq 1$ define the function

$$g_{\omega^j} = g_{\omega_{j-1}} \circ \cdots \circ g_{\omega_0}.$$

Definition 2.4. Let $g_1, \ldots, g_s : \mathbb{R}^n \to \mathbb{R}^n$ be diffeomorphisms and let $Q \subset \mathbb{R}^n$ be compact. The relative global attractor of $g_1, \ldots, g_s$ with respect to Q is defined as

$$A_{Q, g_1, \ldots, g_s} = \bigcap_{\omega \in \Omega} \bigcap_{j \geq 1} g_{\omega^j}(Q) \cap Q. \tag{4}$$

Remark 2.3. Suppose that the g_ℓ are iteration schemes representing the damped Newton method for a function f with different stepsizes. Let Q be a neighborhood of the set of zeros of f. Then, $A_{Q,g_1,\ldots,g_s}$ contains all the zeros of f.

We present now an algorithm for the numerical computation of the relative global attractor $A_{Q,g_1,\ldots,g_s}$, where $g_1,\ldots,g_s$ are diffeomorphisms. Using a multilevel subdivision scheme, this method produces a sequence of sets $\mathcal{B}_0,\mathcal{B}_1,\mathcal{B}_2,\ldots$, where each $\mathcal{B}_k$ consists of finitely many subsets of Q covering the relative global attractor $A_{Q,g_1,\ldots,g_s}$. In the following, we will call the elements B of $\mathcal{B}_k$ boxes. By our construction below,

$$\operatorname{diam}(\mathcal{B}_k)=\max_{B\in\mathcal{B}_k}\operatorname{diam}(B)$$

tends to zero for $k\to\infty$.

In order to guarantee convergence, we have to assume that each dynamical system is applied infinitely often in the subdivision procedure. To make this precise, we choose a sequence $\{u_k\}_{k=1}^{\infty}$ with $u_k\in\{1,\ldots,s\}$ which has the property

$$|\{k:u_k=\ell\}|=\infty,\quad \text{for each } \ell=1,\ldots,s. \tag{5}$$

We describe now the multilevel subdivision procedure in more detail.

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

(i) Subdivision. Construct from $\mathcal{B}_{k-1}$ a new system of subsets $\hat{\mathcal{B}}_k$ such that

$$\bigcup_{B\in\hat{\mathcal{B}}_k}B=\bigcup_{B\in\mathcal{B}_{k-1}}B,$$

$$\operatorname{diam}(\hat{\mathcal{B}}_k)=\theta_k\operatorname{diam}(\mathcal{B}_{k-1}),\quad \text{where } 0<\theta_{\min}\le\theta_k\le\theta_{\max}<1.$$

(ii) Selection. Define the new collection $\mathcal{B}_k$ by

$$\mathcal{B}_k=\left\{B\in\hat{\mathcal{B}}_k:\text{there exists } \hat{B}\in\hat{\mathcal{B}}_k \text{ such that } g_{u_k}^{-1}(B)\cap\hat{B}\neq\emptyset\right\}.$$

Proposition 2.1. See Ref. 10. Let $A_{Q,g_1,\ldots,g_s}$ be the relative global attractor of $g_1,\ldots,g_s$ with respect to $Q\subset\mathbb{R}^n$. Then,

$$\lim_{k\to\infty}h(A_{Q,g_1,\ldots,g_s},Q_k)=0,$$

where $h(B, C)$ denotes the Hausdorff distance between two compact sets $B, C \subset \mathbb{R}^n$.

Example 2.2. Suppose that each g_ℓ represents the damped Newton iteration for a specific steplength h_ℓ; that is,

$$g_\ell(x) = x - h_\ell Df(x)^{-1} f(x),$$

for a certain function $f : \mathbb{R}^n \to \mathbb{R}^n$. Let Q be a neighborhood of the set of zeros $\mathcal{N}_{Q,f}$ of f inside a compact domain in $\mathbb{R}^n$. Then, Proposition 2.1 guarantees that $\mathcal{N}_{Q,f} \subset A_{Q,g_1,\ldots,g_s}$, since the zeros are common fixed points for all the g_ℓ. Moreover, we have equality if Q is chosen to be small enough.

Remark 2.4. The numerical realization of the subdivision procedure works as follows. Rather than working explicitly with the centers and radii of the boxes, these are stored within a binary tree, which leads to a significant reduction in the memory requirement. The selection step is discretized via test points, which are typically chosen within each box in lower dimensions on an a priori specified fixed grid or at random in higher dimensions.

3. Convergence toward Pareto Sets

By Theorem 2.2, each Pareto point is a zero of the function q as defined in the same theorem. In analogy to classical zero-finding procedures with different stepsize strategies, we prove now that an appropriate discretization of the ordinary differential equation (3) will lead to iteration schemes which generate sequences converging toward substationary points. Here, we proceed essentially along the lines of Ref. 15. Then, using the convergence result in Proposition 2.1 we can conclude that an application of the subdivision algorithm yields a close covering of the set of substationary points.

In order to simplify the notation, for a nonnegative vector $b \in [0, 1]^k$, we begin by defining the corresponding objective function [cf. (MOP)]

$$F_b(x) = b^T F(x).$$

We assume that the derivative of each function F_b is Lipschitz continuous with (uniform) Lipschitz constant L.

Now, we discretize (3) and consider the following iteration scheme:

$$x_{j+1} = x_j + h_j p_j, \quad j = 0, 1, \ldots. \tag{6}$$

In the following, sometimes we will denote the right-hand side by P; i.e.,

$$P(x_j) = x_j + h_j p_j, \quad j = 0, 1, \ldots. \tag{7}$$

The descent direction p_j is chosen such that, for positive constants $\sigma, \tau > 0$,

$$-q(x_j)^T p_j / \|q(x_j)\| \|p_j\| \geq \sigma, \quad \|p_j\| \geq \tau \|q(x_j)\|,$$

where $q(x_j)$ is defined in Theorem 2.2. The stepsize h_j is an Armijo or a Powell stepsize. Let x_0 be the initial point and suppose that the iteration can be performed for $j \to \infty$ such that the sequence $(x_j)_{j=0,1,\ldots}$ lies within a compact set $D \subset \mathbb{R}^n$.

Proposition 3.1. Suppose that x^* is an accumulation point of the sequence $(x_j)_{j=0,1,\ldots}$. Then, x^* is a substationary point for the multiobjective optimization problem of Section 2.1.

Proof. First observe that, if one of the x_j is a substationary point, then we are done (cf. Theorem 2.2). Thus, without loss of generality, we may assume that $x_j \to x^*$ for $j \to \infty$ and that none of the x_j is substationary. Consider the corresponding sequence $a_j = (\alpha_1^j, \ldots, \alpha_k^j)$ of solution vectors of the optimization problem of Section 2.1 in step j of the iteration procedure. Since $\alpha_1^j, \ldots, \alpha_k^j \in [0, 1]$, we may assume that $a_j \to a$ for $j \to \infty$. Otherwise, we restrict the following considerations to a subsequence.

We show now that the sequence (x_j) converges to a stationary point for $F_a(x)$, thus proving the desired result.

Using the fact that h_j is an Armijo or a Powell stepsize in x_j, we have by classical results on iteration schemes for optimization problems (see e.g. Ref. 15) that there exists a constant $\theta > 0$ such that

$$F_{a_j}(x_j) - F_{a_j}(x_{j+1}) \geq \theta \min\left[-\nabla F_{a_j}(x_j)^T p_j, \left(\nabla F_{a_j}(x_j)^T p_j / \|p_j\|\right)^2\right] \tag{8}$$

in each step of the iteration process. Observe that θ does not depend on j by the assumption on the uniform Lipschitz continuity of ∇F_b. Now, suppose that

$$F_a(x_j) - F_a(x_{j+1}) < \theta \min\left[-\nabla F_a(x_j)^T p_j, \left(\nabla F_a(x_j)^T p_j / \|p_j\|\right)^2\right] \tag{9}$$

for infinitely many j. By our assumption on the descent direction, we have

$$-\nabla F_a(x_j)^T p_j / \|\nabla F_a(x_j)\| \|p_j\| \geq \sigma/2, \quad \|p_j\| \geq (\tau/2) \|\nabla F_a(x_j)\|,$$

for all $j \geq j_0$. Combining these estimates with (8) and (9) we obtain

$$\begin{aligned} 0 &= \lim_{j\to\infty} \min\left[-\nabla F_a(x_j)^T p_j, \left(\nabla F_a(x_j)^T p_j/\|p_j\|\right)^2\right] \\ &\geq (\theta\sigma/4)\min(\tau,\sigma)\|\nabla F_a(x^*)\|^2 \end{aligned}$$

and x^* is substationary as desired.

It remains to consider the case where

$$F_a(x_j) - F_a(x_{j+1}) \geq \theta \min\left[-\nabla F_a(x_j)^T p_j, \left(\nabla F_a(x_j)^T p_j/\|p_j\|\right)^2\right],$$

for all $j \geq j_1$. Here, we obtain in an analogous way

$$F_a(x_j) - F_a(x_{j+1}) \geq (\theta\sigma/4)\min(\tau,\sigma)\|\nabla F_a(x_j)\|^2 \geq 0,$$

for all $j \geq \max(j_0, j_1)$. Letting $j \to \infty$, it follows that

$$\|\nabla F_a(x^*)\| = 0. \qquad \square$$

For the remainder of this section, we assume now that $p_j = q(x_j)$ and that we are working with s different stepsizes h_ℓ. Then, combining Propositions 3.1 and 2.1, we obtain immediately the following result.

Corollary 3.1. Suppose that the set $\mathcal{S}$ of substationary points is bounded and let D be a compact neighborhood of $\mathcal{S}$. Then, an application of the subdivision algorithm to D with respect to the iteration scheme (6) creates a covering of the entire set $\mathcal{S}$; that is,

$$\mathcal{S} \subset Q_k, \quad \text{for } k = 0, 1, 2, \ldots,$$

in the course of the subdivision processes.

Observe that we have shown that the covering obtained by the subdivision process becomes tight. However, we cannot prove convergence toward $\mathcal{S}$ without an additional assumption on its structure as the following example shows.

Example 3.1. Let us reconsider Example 2.1 (cf. Fig. 1). In that case, an application of the subdivision algorithm to the interval $D = [-1, 3]$ will converge to the interval $[0, 2]$. Thus, we obtain a covering of the set $\mathcal{S} = [0, 1] \cup [1.5, 2]$, but we also approximate the additional part (1, 1.5).

For proof of this fact, one has to observe that a box which contains the number 1 as well as points which are bigger than 1 has always a

nonzero intersection with its image under the iteration scheme (6). Moreover, the image of this box has also a nonzero intersection with its right neighbor. Proceeding with this neighboring box, we see that all the boxes between 1 and 1.5 have preimages in other boxes in each step of the subdivision process. Therefore, the interval (1, 1.5) is never removed in the selection step.

We will see in Section 4 how to overcome the problem described in the previous example in actual realizations of the algorithm. However, these considerations in combination with standard compactness arguments lead immediately to the following convergence result.

Corollary 3.2. suppose that the set $\mathcal{S}$ of substationary points is bounded and connected. Let D be a compact neighborhood of $\mathcal{S}$. Then, an application of the subdivision algorithm to D with respect to the iteration scheme (6) leads to a sequence of coverings which converges to the entire set $\mathcal{S}$; that is,

$$h(\mathcal{S}, Q_k) \to 0, \quad \text{for } k = 0, 1, 2, \ldots,$$

where h denotes the Hausdorff distance.

Remark 3.1. Under the assumptions of Corollary 3.2, we can conclude that in the unconstrained case the set $\mathcal{S}$ has to have a trivial homotopy. For instance, in the case of two objective functions on (at least) a two-dimensional parameter space, the set $\mathcal{S}$ cannot be topologically equivalent to a circle.

4. Algorithms, Realizations, and Extensions

In this section, we propose three different algorithms for the computation of the Pareto set of a given (MOP); or to be more precise, we present algorithms for the computation of tight coverings of such sets. Moreover, we propose some guidelines on how to combine these algorithms in order to improve the performance of the respective numerical schemes.

4.1. Subdivision Algorithm. The first algorithm is based directly on the theoretical considerations of the previous section, in particular on Corollaries 3.1 and 3.2. In fact, we discuss now a concrete realization of the subdivision procedure for the computation of tight coverings of the set of substationary points using the dynamical system

$$x_{j+1} = P(x_j) = x_j + h_j p_j, \quad j = 0, 1, \ldots.$$

Descent Direction. In all the computations presented in this paper, we have used the descent direction $p_j = q(x_j)$; cf. (6) and Theorem 2.2.

For the specific case of bicriteria optimization problems (i.e., $k = 2$), one could use alternatively the following direction:

$$p_j = -\left(\nabla f_1(x_j)/||\nabla f_1(x_j)||_2 + \nabla f_2(x_j)/||\nabla f_2(x_j)||_2\right).$$

This choice is particularly useful in the case where the cost for the evaluation of $\nabla f_i, i = 1, 2$, is high. We have tested this descent direction with all the bicriteria optimization problems presented in this article yielding satisfying results.

Steplength. Following standard techniques for steplength control (see e.g. Ref. 15), we have chosen a particular Armijo stepsize strategy in the following way: starting with the given point x_j, we evaluate F along the descent direction p_j in uniform steplengths h_0 as long as the values of all the objectives decrease. Once one objective function starts to increase, a better iterate x_{j+1} with intermediate step length is calculated via backtracking.

Algorithm 4.1.

Step 1. Determine the largest $n \in \mathbb{N}$ such that, for every $i = 1, \ldots, k$,

$$F(x_j + nh_0 p_j) <_p F(x_j + (n-1)h_0 p_j),$$
$$\langle \nabla f_i(x_j + (n-1)h_0 p_j),\, p_j \rangle < 0.$$

Step 2. Choose $x_{j+1} \in [x_j + (n-1)h_0 p_j,\, x_j + nh_0 p_j]$ such that

$$F(x_{j+1}) <_p F(x_j + (n-1)h_0 p_j).$$

Remark 4.1.

(a) To find an appropriate guess for the scanlength h_0, it is possible to take advantage of the subdivision scheme. If a steplength $\tilde{h}$ has been computed for a point $\tilde{x}$ inside a certain box, then this distance can be chosen as the scan length for the following points inside the same box. This strategy works particularly well when the subdivision scheme is at a level where all the boxes are already quite small.

(b) Concretely, we propose the following backtracking in Step (2) above. For every $i \in \{i, \ldots, k\}$ with $f_i(x_j + nh_0 p_j) > f_i(x_j + (n-1)h_0 p_j)$, we determine

$$x_j^i = x_j + ((n-1)h_0 + \Theta_i)p_j, \quad \Theta_i \in (0, 1),$$

via quadratic backtracking. Then we set $\hat{x}_j = x_j^i$, where i is chosen such that Θ_i is minimal. If

$$F(\hat{x}_j) <_p F(x_j + (n-1)h_0 p_j),$$

then the point $\hat{x}_j$ is accepted and we choose $x_{j+1} = \hat{x}_j$ in Step 2. Otherwise we proceed in the same way to find a new iterate between $x_j + (n-1)h_0 p_j$ and $\hat{x}_j$.

4.2. Recovering Algorithm. It may be the case that, in the course of the subdivision procedure, boxes get lost although they contain substationary points. For instance, this will be the case when there are not enough test points taken into account for the evaluation of $F(B)$ for a box $B \in \mathcal{B}_k$ (see Remark 2.4). We describe now an algorithm using a kind of healing process, which allows us to recover those substationary points which have previously been lost.

Before we can state the algorithm, we have to present some more technical details about box collections. For theoretical purposes, denote by $\mathcal{P}_k$ a complete partition of the set $Q = B_{\hat{c},\hat{r}}$ into boxes[5] of subdivision size or depth k, which are generated by successive bisection of Q. Then, for every point $y \in Q$ and every depth k, there exists exactly one box $B(y,k) \in \mathcal{P}_k$ with center c and radius r such that

$$c_i - r_i \le y_i < c + r_i, \quad \forall i = 1, \ldots, n.$$

The aim of the algorithm is to extend the given box collection step-by-step along the covered parts of the set $\mathcal{S}$ of substationary points until no more boxes are added. In order to find the corresponding neighboring boxes of a box B, we take starting points $\{s_i\}_{i=1,\ldots,l}$ near B and compute $\mathcal{X} = \{P^q(s_i) | i = 1, \ldots, l\}$ with a suitable power $q \ge 1$ in order to obtain points both near B and $\mathcal{S}$. Afterward, the box collection is extended by the boxes $B \in \mathcal{P}_k$ which contain elements from $\mathcal{X}$. In the first step, this is done for all the boxes from the box collection; for the following steps, this local search has to be performed only in the neighborhood of the boxes which were added in the preceding step (see Fig. 2).

For a given box collection $\mathcal{B}_k$, the algorithm reads as follows:

Algorithm 4.2.

Step 1. Mark all boxes $B \in \mathcal{B}_k$.

[5] $\mathcal{P}_k$ and hence every box collection considered here can be identified with a set of leaves of a binary tree of depth k.

Step 2. For all marked $B \in \mathcal{B}_k$, unmark a box and choose starting points $\{s_i\}_{i=1,\dots,l}$ near B; compute $\mathcal{X} := \{P^q(s_i) | i = 1, \dots, l\}$. For all $y \in \mathcal{X}$, if $B(y,k) \notin \mathcal{B}_k$ add B(y,k) to the collection $\mathcal{B}_k$ and mark the box.
Repeat Step 2 while new boxes are added to $\mathcal{B}_k$ or until a prescribed maximal number or steps is reached.

Hence, the recovering algorithm allows us to add boxes to the given collection. The desired covering of the set $\mathcal{S}$ of substationary points cannot get worse, but will improve if the parameters of the algorithm are adjusted properly. On the other hand, the recovering algorithm does not perform adequately in the case where a box does not contain part of $\mathcal{S}$, but is possibly far away. In this case, the algorithm would extend the box covering by many undesired regions on the way toward $\mathcal{S}$ in the course of the iteration of test points. This observation is particularly valid for higher-dimensional parameter spaces. A method which allows to overcome this problem can be found in Ref. 16.

4.3. Sampling Algorithm. Observe that there are some potential drawbacks which may occur when using the two algorithms described above:

(a) The gradients of the objectives are needed.
(b) The set $\mathcal{S}$ is generally a strict superset of the Pareto set.
(c) The algorithms are capable of finding local Pareto points on the boundary of the domain Q, e.g. via penalization strategies. However, it turned out in practice that typically (MOP)s contain many local Pareto points on ∂Q which are not globally optimal.

The following sampling algorithm avoids all these problems because it takes into account only the values of the objective functions. On the other hand, this algorithm is not as robust wrt errors as the first two algorithms, because it is only global relative to the underlying box collection.

In the following, we call a point $x \in \mathcal{X} \subset \mathbb{R}^n$ nondominated with respect to F and $\mathcal{X}$ if there does not exist a point $y \in \mathcal{X}$ with $F(y) \leq_p F(x)$ and $F(y) \neq F(x)$. Using this notion, an outline of the algorithm is as follows. Given a box collection $\mathcal{B}_{k-1}$, the collection $\mathcal{B}_k$ is obtained via the following algorithm.

Algorithm 4.3.
Step 1. Subdivision. This is as in Section 2.2.
Step 2. Selection. For all $B \in \hat{\mathcal{B}}_k$, choose a set of test points $\mathcal{X}_B \subset B$,

$$\text{N:= nondominated points of } \bigcup_{B \in \hat{B}_k} \mathcal{X}_B,$$

$$\mathcal{B}_k := \{B \in \hat{\mathcal{B}}_k : \exists y \in \mathcal{X}_B \cap N\}.$$

The approach of this algorithm is very close in spirit to well-known branch and bound strategies used for scalar optimization, e.g. described in Ref. 11 or in Refs. 17–18.
But in contrast to these algorithms, we have omitted any bounding strategy. This can be done because of the special structure of the problem: the larger the number k of objectives is, the more robust the selection step of the algorithm becomes. These observations coincide with our intuition concerning the nature of multiobjective optimization. For the efficient realization of the nondominance test in the selection step, we use basically the data structure described in Ref. 19.

4.4. Usage and Combination of the Algorithms. In principle, each of the algorithms proposed above is applicable to an (MOP) on its own. The subdivision algorithm has the advantage of being very robust with respect to errors by the use of the descent direction. On the other hand, all the gradients of the objectives have to be available and the algorithm is unable to distinguish between a local and a global Pareto point. Furthermore, the efficiency of the algorithm will get worse when the (MOP) has optima relative to the boundary of the domain. The recovering algorithm is able to extend the computed box covering of the set of substationary points, but is just local in nature. The sampling algorithm is able to detect global Pareto points even on the boundary of the domain, due to the fact that it works in the image space of the (MOP). However, there remains always uncertainty because of the sampling approach, in particular when the boxes are big or the dimension of the (MOP) is large. Nevertheless, it turned out in practice that this algorithm works quite well, in particular when the gradients of the objectives are not available and the dimension of the (MOP) is moderate.

To obtain a better performance, i.e. to compute a robust approximation of the Pareto set and to use a moderate amount of function calls, we propose the following combination of the three algorithms.

Step 1. Start with the subdivision algorithm because of its robustness. Take a few test points for the evaluation of the boxes via the dynamical system P.

Step 2. Apply the recovering algorithm to the box collection which has been computed in Step 1. This fills the gaps which have possibly been generated in Step 1.

Step 3. Use the sampling algorithm to tighten the extended box collection. By using this algorithm, boxes which contain only local Pareto points can be removed from the covering. Furthermore, boxes get removed which contain no substationary points, but where kept in Step 1 because of the weak convergence of P in these regions.

There are of course other possible ways to combine the algorithms; e.g., it is possible to apply again the subdivision algorithm on the box collection obtained by the procedure described above. In Step 2, the number of boxes which are added to the collection is a measure for the number of test points needed in Step 1.

5. Numerical Results

In this section, we illustrate the efficiency of our set-oriented algorithms via several numerical examples.

Example 5.1. In this example, we consider three objective functions $f_1, f_2, f_3 : \mathbb{R}^3 \to \mathbb{R}$,

$$f_1(x_1, x_2, x_3) = (x_1 - 1)^4 + (x_2 - 1)^2 + (x_3 - 1)^2,$$
$$f_2(x_1, x_2, x_3) = (x_1 + 1)^2 + (x_2 + 1)^4 + (x_3 + 1)^2,$$
$$f_3(x_1, x_2, x_3) = (x_1 - 1)^2 + (x_2 + 1)^2 + (x_3 - 1)^4.$$

The basic domain is chosen as $Q = [-5, 5]^3$. The resulting box collections are shown in Fig. 3 a–c. Here, we have taken a $3 \times 3 \times 3$ grid as test points for every box.

Alternatively, we have used a combination of the three different algorithms to achieve a better performance. The result shown in Fig. 3 d–f was obtained by the following steps: first, the subdivision algorithm was applied for 21 steps using only the center point of every box as the test point for dynamical system P (Fig. 3d). Using only these few test points, the computed box collection $\mathcal{B}_{21}$ reveals already the shape of the set of Pareto points, but it contains also many holes. These holes could be filled by an application of the recovering algorithm on $\mathcal{B}_{21}$ (Fig. 5e). Finally, the covering was tightened using the sampling algorithm.

Example 5.2. We solve now an (MOP) which serves as a model for a problem occuring in production planning (cf. Ref. 7). Here, we have two

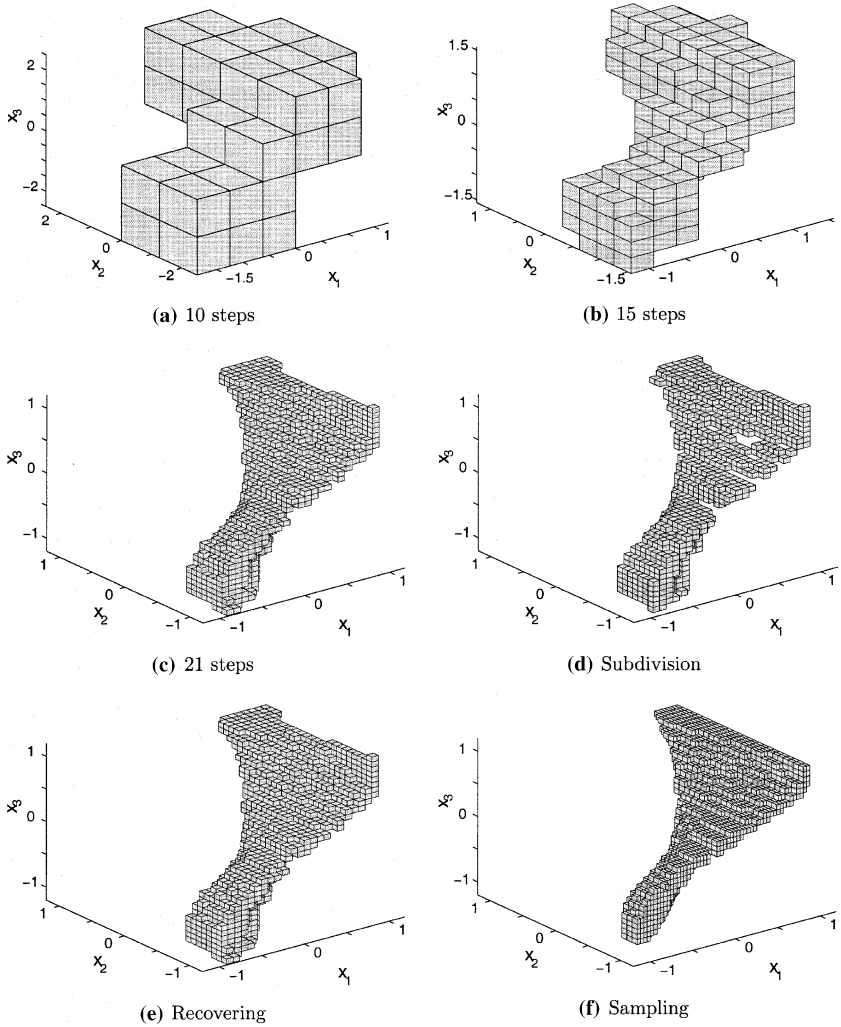

Fig. 3. Results for Example 5.1. (a)–(c) Resulting box collections using 27 test points per box after 10, 15, and 25 steps. (d)–(f) Combination of the three algorithms.

objective functions $f_1, f_2 : \mathbb{R}^n \to \mathbb{R}$,

$$f_1(x) = \sum_{j=1}^{n} x_j, \quad f_2(x) = 1 - \prod_{j=1}^{n} x_j (1 - w_j(x_j)),$$

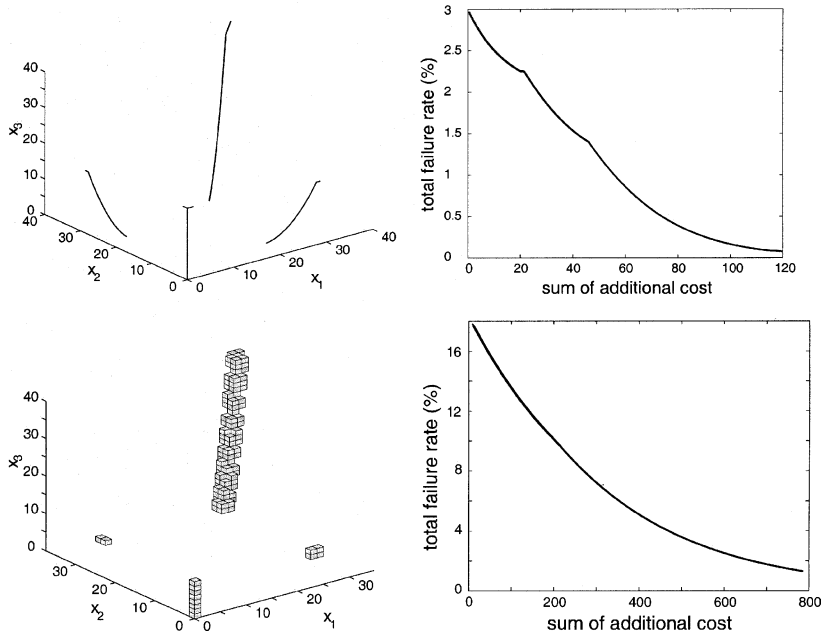

Fig. 4. Results for Example 5.2. For dimension $n = 3$ (top) after 30 iterations and for dimension $n = 20$ (bottom) after 100 iterations in parameter space (left) and in image space (right).

where

$$w_j(z) = \begin{cases} 0.01 \cdot \exp(-(z/20)^{2.5}), & j = 1, 2, \\ 0.01 \cdot \exp(-(z/15), & 3 \le j \le n. \end{cases}$$

The two objective functions have to be interpreted as follows. The function f_1 represents the sum of the additional cost necessary for a more reliable production of n items; these items are needed for the composition of a certain product. The function f_2 describes the total failure rate for the production of this composed product. The basic domain is $Q = [0, 40]^n$. For $n = 3$ and $n = 20$, the approximations are shown in Fig. 4. These were obtained by the sampling algorithm, which is capable to detect Pareto optimal solutions on the boundary of the domain Q.

Example 5.3. Optimization of an Active Suspension. In this section, we illustrate that the developed algorithms can be useful in applications to

real world problems. The following example of optimizing an active suspension is taken from the field of automotive engineering. As it is not the aim of this paper to focus on suspensions, only a reduced problem will be dealt with.

In suspension engineering, it is common to analyze the basic principles of the active suspension regarding the system set-up and controller design by using quarter car models. These models consist of the proportional mass of the car body, one wheel, and the respective strut (e.g. Ref. 20). Due to energy considerations, the active intervention of the suspension is restricted usually to lower frequencies (e.g. Ref. 21), which are in good approximation characterized by the dynamical behavior of the car body only. Therefore the basic design of the active suspension can be based on a simple model: The road z_0 excites the mass m_B with its coordinate z_B via the strut with spring constant c_{strut} and damping constant d_{strut}. The active suspension generates an active force F_{active} on m_B. Using acceleration and level measurements, the car body dynamics can be attuned to the transfer function

$$G(\mathrm{s}) = \frac{z_B}{z_0}(\mathrm{s}) = \frac{d\mathrm{s} + c}{m_B \mathrm{s}^2 + (d + d_s)\mathrm{s} + c}. \tag{10}$$

The spring constant c and the damper constant d are composed of the physical constants c_{strut} and d_{strut} respectively and an additional controller part using the relative velocity between the road and car body. The additional damper term $d_s\mathrm{s}$ in the denominator can be obtained by using the absolute car body velocity derived from a measurement of the acceleration.

The fundamental design problem of (10) can be seen easily when looking at a step and a ramp response (Fig 5a, see also Ref. 22). The left-hand side shows the response of the car body to a step on the road of height 1 cm. The spring constant c is kept constant; d and d_s are varied such that $d + d_s = \text{const}$. With increasing d_s, the overshoot decreases, which means that the driving comfort increases. At the same time, it becomes difficult to move up a ramp (Fig. 5a, right-had side). The ramp excitation corresponds to a drive with 20 m/sec on a 15% slope. When the damping of the system is realized by d_s only ($d = 0$), the simulation shows a ramp error of 57cm, which means that the car hits the mechanical buffers while driving along the slope. Thus, in order to optimize the suspension behavior, it is apparently necessary to consider not only a comfort criterion but also the ramp error.

Driving tests show that a body response of 1 Hz and a damping of 0.6 are perceived to be particularly comfortable. This suggests (11) as a

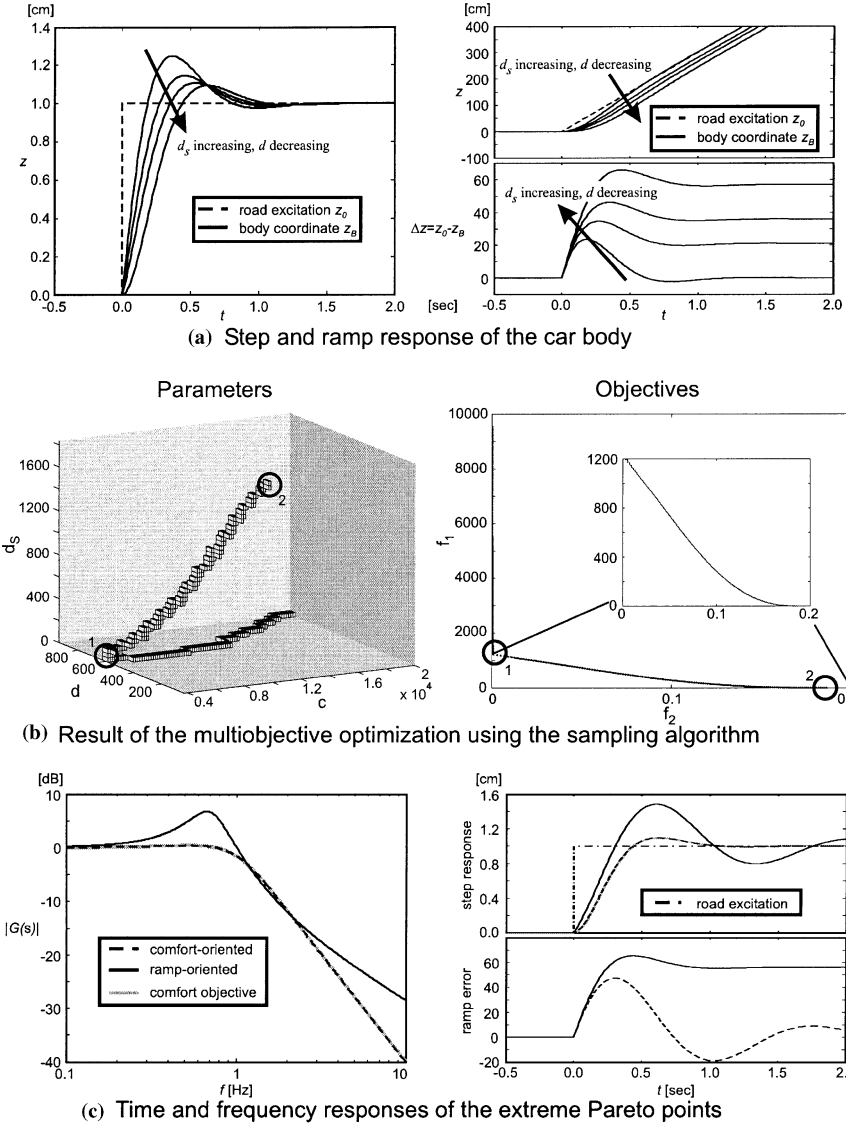

Fig. 5. Results for Example 5.3. Optimization of an active car suspension.

reference car body response with respect to comfort.

$$G_{\text{c.o.}}(s) = \frac{1}{T_{\text{c.o.}} s^2 + 2 d_{\text{c.o.}} s + 1}, \quad T_{\text{c.o.}} = (1/2\pi)\text{rad/sec}, d_{\text{c.o.}} = 0.6. \quad (11)$$

The difference between (10) and (11) can be used as a criterion for comfort. As both functions are minimum phase, it is sufficient to compare the magnitudes of the transfer functions. These considerations lead to the first objective function

$$f_1 = \sum_{j=0,\ldots,450} \left[20\log_{10} \left| G\left(i10^{-3+4\cdot 5\frac{j}{450}} \right) \right| \right.$$
$$\left. -20\log_{10} \left| G_{\text{c.o.}}\left(i10^{-3+4\cdot 5\frac{j}{450}} \right) \right| \right]^2. \tag{12}$$

As we have already seen in Fig. 5a, it is necessary to add another criterion for the ramp error. Computing the ramp error for $t \to \infty$, we obtain the second objective function

$$f_2 = \lim_{\mathrm{s}\to 0} \mathrm{s}(1 - G(\mathrm{s}))\frac{1}{\mathrm{s}^2} = \frac{d_{\mathrm{s}}}{c}. \tag{13}$$

Here, c, d, d_s are the free parameters for the optimization, m_B is set to 250 kg. All parameters must be positive to avoid rhs poles and zeros.

The Pareto set computed by multiobjective optimization (Fig. 5b) can be divided into two parts: the upper part in parameter space with $d_s > 0$ belongs to the right-hand part of the objectives diagram in Fig. 5b. The second part in parameter space lies in the plane where $d_s = 0$. The corresponding curve in image space has an extremely steep gradient. However, from a physical point of view, this branch is irrelevant. The extremal substationary points of the multiobjective optimization problem are therefore the points 1 and 2 (see the left-hand side in Fig. 5b). The comfort-oriented substationary point 2 yields the prescribed comfort optimum (11), which can be seen both in the frequency domain and in the time domain (Fig. 5c). The ramp-oriented substationary point 1 returns a ramp error of 0 (suggested only by the ramp error in Fig. 5c, but it can clearly be seen in Fig. 5b). However, with 0.26, the damping of this system is unnecessarily low. This is due to the choice of the comfort criterion f_1. The ramp-oriented frequency response in Fig. 5c illustrates the reason: its deviation from the comfort optimum around its peak response is weighed in the same way as the deviation at higher frequencies far below 0dB gain. An increase of damping using d instead of d_s (and thus improving damping without deteriorating the ramp error) leads to a lower peak-response, but also to higher gain at higher frequencies. This problem could be dealt with by penalty functions.

The optimized set of parameter values determined in the way presented here allows for a controller tuning according to the target customer

group. The suspension design process can be simplified, as test drivers just need to switch between Pareto points and thus do not have to tune all free parameters without additional support. In addition, multiobjective optimization of suspension controllers also has the potential for further suspension improvements. For instance, in Ref. 22, it is shown that the Pareto set can be used for the self-optimization of a car.

6. Conclusions

In this article, we present new set-oriented algorithms for the solution of global multiobjective optimization problems. We have analyzed their convergence properties and we have demonstrated their efficiency by several numerical examples. Due to the set-oriented approach, the algorithms are robust and they are able to approximate the entire set of global Pareto points within a compact domain. Furthermore, they can in principle be applied to both smooth and nondifferentiable objectives. On the other hand, the computational time is quite large in higher- dimensional parameter space. We expect to overcome this problem by a construction of more efficient iteration schemes. Finally, the modification of these techniques for an application to discrete (MOP)s will be another challenge for further investigations.

References

1. Miettinen, K., *Nonlinear Multiobjective Optimization*, Kluwer Academic Publishers, Boston, Massachusetts, 1999.
2. Deb, K., *Multiobjective Optimization Using Evolutionary Algorithms*, Wiley, New York, NY, 2001.
3. Das, I., and Dennis, J., *Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems*., SIAM Journal on Optimization, Vol. 8, pp. 631–657, 1998.
4. Zitzler, E., Deb, K., Thiele, L., Coello Coello, C. A., and Corne, D., *Evolutionary Multicriterion Optimization*, 1st International Conference, EMO 2001, Springer, Berlin, Germany, 2001.
5. Fonseca, S. M., Fleming, P. J., Zitzler, E., Deb, K., and Thiele, L., *Evolutionary Multicriterion Optimization*, 2nd International Conference, EMO 2003, Springer, Berlin, Germany, 2003.
6. Hillermeier, C., *Nonlinear Multiobjective Optimization : A Generalized Homotopy Approach*, Birkhäuser, Basel, Switzerland, 2001.
7. Schäffler, S., Schultz, R., and Weinzierl, K., *A Stochastic Method for the Solution of Unconstrained Vector Optimization Problems*, Journal of Optimization Theory and Applications, Vol. 114, pp. 209–222, 2002.

8. Dellnitz, M., and Hohmann, A., *A Subdivision Algorithm for the Computation of Unstable Manifolds and Global Attractors*, Numerische Mathematik, Vol. 75, pp. 293–317, 1997.
9. Dellnitz, M., and Hohmann, A., *The Computation of Unstable Manifolds Using Subdivision and Continuation*, Nonlinear Dynamical Systems and Chaos, Edited by H. W. Broer, S. A. Van Gils, I. Hoveijn, and F. Takens, Birkhäuser, Basel, Switzerland, Vol. 19, pp. 449–459, 1996.
10. Dellnitz, M., Schütze, O., and Sertl, S., *Finding Zeros by Multilevel Subdivision Techniques*, IMA Journal of Numerical Analysis, Vol. 22, pp. 167–185, 2002.
11. Horst, R., and Tuy, H., *Global Optimization: Deterministic Approaches*. Springer, Berlin, Germany, 1993.
12. Pareto, V., *Cours d'Economie Politique*, Libraire Droz, Genève, Switzerland, 1964 (First Edition in 1896).
13. Kuhn, H., and Tucker, A., *Nonlinear Programming*, Proceedings of the Berkeley Symposium on Mathematical and Statistical Probability, Edited by J. Neumann, University of California at Berkeley, Berkeley, California, pp. 481–492, 1951.
14. Junge, O., *Mengenorientierte Methoden zur numerischen Analyse dynamischer Systeme*, PhD Thesis, University of Paderborn, 1999.
15. Dennis, J. E., and Schnabel, R. B., *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, New Jersey, 1983.
16. Schütze, O., Mostaghim, S., Dellnitz, M., and Teich, J., *Covering Pareto Sets by Multilevel Evolutionary Subdivision Techniques*, Evolutionary Multicriterion Optimization, Edited by C. M. Fonseca, P. J. Fleming, E. Zitzler, D. Deb, and L. Thiele, Springer, Berlin, Germany, Vol. 2, pp. 118–132, 2003.
17. Sertl, S., and Dellnitz, M., *Global Optimization Using a Dynamical Systems Approach*, Journal of Global Optimization (submitted).
18. Chauduri, I., Sertl, S., Zoltán, H., Dellnitz, M., and Fraunheim, T., *Global Optimization of Silicon Nanoclusters*, Applied Surface Science (submitted).
19. Schütze, O., *A New Data Structure for the Nondominance Problem in Multiobjective Optimization*, Evolutionary Multicriterion Optimization, Edited by C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, Springer, Berlin, Germany, Vol. 2, pp. 509–518, 2003.
20. Mitschke, M., *Dynamik der Kraftfahrzeuge, Volume C: Fahrverhalten*, 2nd Edition, Springer, Berlin, Germany, 1990.
21. Castiglioni, G., Jäker, K., and Schlüter, F., *Das aktive Fahrwerk mit elektrischen Aktuatoren*, AT Automatisierungstechnik, Vol. 7, pp. 345–350, 1996.
22. Hestermeyer, T., and Oberschelp, O., *Selbstoptimierende Fahrzeugregelung- Verhaltensbasierte Adaption*, Intelligente mechatronische Systeme, HNI-Verlagsschriftenreihe, Heinz Nixdorf Institut, Vol. 122, pp. 231–241, 2003.