A survey of recent trends in multiobjective optimal control – surrogate models, feedback control and objective reduction

Sebastian Peitz[1],*, and Michael Dellnitz[1]

[1] Paderborn University, Warburger Str. 100, 33100 Paderborn, Germany
* Correspondence: speitz@math.upb.de

Abstract: Multiobjective optimization plays an increasingly important role in modern applications, where several criteria are often of equal importance. The task in multiobjective optimization and multiobjective optimal control is therefore to compute the set of optimal compromises (the *Pareto set*) between the conflicting objectives. The advances in algorithms and the increasing interest in Pareto optimal solutions have led to a wide range of new applications related to optimal and feedback control which results in new challenges such as expensive models or real-time applicability. Since the Pareto set generally consists of an infinite number of solutions, the computational effort can quickly become challenging which is particularly problematic when the objectives are costly to evaluate or when a solution has to be presented very quickly. This article gives an overview over recent developments in accelerating multiobjective optimal control for complex problems where either PDE constraints are present or where a feedback behavior has to be achieved. In the first case, surrogate models yield significant speed-ups. Besides classical meta-modeling techniques for multiobjective optimization, a promising alternative for control problems is to introduce a surrogate model for the system dynamics. In the case of real-time requirements, various promising model predictive control approaches have been proposed, using either fast online solvers or offline-online decomposition. We also briefly comment on dimension reduction in many-objective optimization problems as another technique for reducing the numerical effort.

Keywords: Multiobjective Optimization; Optimal Control; Model Order Reduction; Model Predictive Control

MSC: 90C29;49M37

---

1. Introduction

There is hardly ever a situation where only one goal is of interest at the same time. When performing a purchase for example, we want to pay a low price while getting a high quality product. In the same manner, multiple goals are present in most technical applications, maximizing quality versus minimizing the cost being only one of many examples. This dilemma leads to the field of *multiobjective optimization*, where we want to optimize all relevant objectives simultaneously. However, this is obviously impossible as the above example illustrates. Generically, the different objectives contradict each other such that we are forced to choose a compromise. While we are usually satisfied with one optimal solution in the scalar valued setting, there exists in general an infinite number of *optimal compromises* in the situation where multiple objectives are present. The set of these compromise solutions is called the *Pareto set*, the corresponding points in the objective space form the *Pareto front*.

Since the solution to a multiobjective optimization problem (MOP) is a set, it is significantly more expensive to compute than the optimum of a single objective problem, and many researchers devote their work to the development of algorithms for the efficient numerical approximation of Pareto sets. These advances have opened up new challenging application areas for multiobjective optimization. In optimal control, the optimization variable is not finite-dimensional but a rather function, typically depending on time. The goal is to steer a dynamical system in such a way that one (ore multiple)

objectives are minimized. Two particularly challenging control problems are feedback control and control problems constrained by partial differential equations (PDEs). In the first case, the time for computing the Pareto set is strictly limited, often to a small fraction of a second. In the latter case, even the solution of single objective problems is often extremely time consuming so that the development of new algorithmic ideas is necessary to make these problems computationally feasible.

In fact, in situations like these, surrogate models or dimension reduction techniques are a promising approach for significantly reducing the computational effort and thereby enabling real-time applicability. This article gives an overview over recent advances in surrogate modeling for multiobjective optimal control problems, where the approach is to replace the underlying system dynamics by a reduced order model which can be solved much faster. On the other hand, it introduces an approximation error which has to be taken into account when analyzing convergence properties. The article is structured as follows. In Section 2, we are going to review some basics about multiobjective optimization, including the most popular solution methods. The two main challenges are addressed individually in the next sections, starting with expensive models in Section 3. There are also surrogate modeling approaches for MOPs which directly provide a mapping from the control variable to the objective function values. Since there already exist extensive surveys for this case (cf. [1–3], for instance), these are summarized only very briefly. In Section 4, real-time feedback control is discussed. Finally, we briefly discuss the question of dimension reduction in the number of objectives in Section 5 before concluding with a summary of further research directions in Section 6.

2. Multiobjective optimization

In this section the concepts of multiobjective optimization and Pareto optimality will be introduced and some widely-used solution methods will be summarized. More detailed introductions to multiobjective optimization can be found in, e.g., [4,5].

2.1. Theory

In multiobjective optimization we want to minimize multiple objectives at the same time. Consequently, the fundamental difference to scalar optimization is that the objective function $J : \mathcal{U} \to \mathbb{R}^k$ is vector-valued. Hence, the general problem is of the form

$$\min_{u \in \mathcal{U}} J(u) = \min_{u \in \mathcal{U}} \begin{pmatrix} J_1(u) \\ \vdots \\ J_k(u) \end{pmatrix} \tag{MOP}$$
$$\text{s.t.} \quad g_i(u) \leq 0, \quad i = 1, \ldots, l,$$
$$\quad\quad h_j(u) = 0, \quad j = 1, \ldots, m,$$

where $u \in \mathcal{U}$ is the *control variable* and $g : \mathcal{U} \to \mathbb{R}^l$, $g(u) = (g_1(u), \ldots, g_l(u))^\top$ and $h : \mathcal{U} \to \mathbb{R}^m$, $h(u) = (h_1(u), \ldots, h_m(u))^\top$, are inequality and equality constraints, respectively. The space of the control variables $\mathcal{U}$ is also called the *decision space* (according to the term *decision variable* for u in classical multiobjective optimization) and the objective function maps u to the *objective space*. Depending on the problem setup, $\mathcal{U}$ can either be finite-dimensional, i.e. $\mathcal{U} = \mathbb{R}^n$, or some appropriate function space.

Remark 1. *It is common in finite-dimensional optimization to use the notation x for the control or optimization variable and F for the objective function. In contrast to that, u and J are more common for control problems. In order to unify the notation, the latter will be used throughout this article for all optimization and optimal control problems.*

In contrast to classical optimization, in optimal control we have to compute an input in such a way that a dynamical system behaves optimally with respect to some specified cost functional. Hence,

we have the system dynamics as an additional constraint, very often in the form of ordinary (ODEs) or partial differential equations (PDEs):

$$\dot{y}(x,t) = G(y(x,t), u(t)), \quad (x,t) \in \Omega \times (t_0, t_e],$$

$$a(x,t)\frac{\partial y}{\partial n}(x,t) + b(x,t)y(x,t) = c(x,t), \qquad (x,t) \in \Gamma \times (t_0, t_e], \qquad \text{(PDE)}$$

$$y(x,t_0) = y_0(x), \qquad\qquad x \in \Omega,$$

where the domain of interest $\Omega \subset \mathbb{R}^{n_x}$ is a connected open set with spatial dimension n_x and the boundary is denoted by $\Gamma = \partial\Omega$ with outward normal vector n. The coefficients $a(x,t)$, $b(x,t)$ and $c(x,t)$ in the boundary condition are given by the problem definition. The operator G is a partial differential operator describing the evolution of the system. The cost functional $\hat{J} : \mathcal{U} \times \mathcal{Y} \to \mathbb{R}^k$ of an optimal control problem consequently depends on the control u as well as the system state y, which results in a *multiobjective optimal control problem*:

$$\min_{u \in \mathcal{U}, y \in \mathcal{Y}} \hat{J}(u,y) = \min_{u \in \mathcal{U}, y \in \mathcal{Y}} \begin{pmatrix} \int_{t_0}^{t_e} C_1(y(x,t), u(t))\,dt + \Phi_1(y(x,t_e)) \\ \vdots \\ \int_{t_0}^{t_e} C_k(y(x,t), u(t))\,dt + \Phi_k(y(x,t_e)) \end{pmatrix}$$

$$\text{s.t.} \qquad \text{(PDE)}$$

$$g_i(y,u) \le 0, \quad i = 1, \dots, l,$$

$$h_j(y,u) = 0, \quad j = 1, \dots, m.$$

(MOCP)

There are articles on multiobjective optimal control which specifically address the implications of multiple objectives for optimal control [6,7], see also [8] for a short survey of methods. However, for many problems there exists a unique solution y for every u such that (MOCP) can be simplified by introducing a so-called *control-to-state operator* $\mathcal{S} : \mathcal{U} \to \mathcal{Y}$, see [9] for details. By setting $\hat{J}(u,y) = \hat{J}(u, \mathcal{S}u) = J(u)$, the problem is transformed into (MOP). For this reason, we will from now on only consider (MOP).

In the situation where $\mathcal{U}$ is a function space (i.e., in the case of optimal control), the problem can be numerically transformed into a high- yet finite-dimensional problem in a *direct solution method* via discretization, cf. [10,11]. This results in a large number of control variables which can be very challenging on its own in multiobjective optimization. If the system dynamics are governed by a PDE, then the spatial discretization of the state y results in an even higher number of unknowns which can easily reach several millions or more [12].

In contrast to single objective optimization problems, there exists no total order of the objective function values in $\mathbb{R}^k$ with $k \ge 2$ (unless they are not conflicting). Therefore, the comparison of values is defined in the following way [4]:

Definition 1. *Let $v, w \in \mathbb{R}^k$. The vector v is* less than w *(denoted by $v < w$), if $v_i < w_i$ for all $i \in \{1, \dots, k\}$. The relation $\le$ is defined in an analogous way.*

A consequence of the lack of a total order is that we cannot expect to find isolated optimal points. Instead, the solution to (MOP) is the set of optimal compromises (also called the *Pareto set* or *set of non-dominated points*):

Definition 2. *Consider the multiobjective optimization problem* (MOP)*. Then*

1. *a point u^** dominates *a point u, if $J(u^*) \le J(u)$ and $J(u^*) \ne J(u)$.*
2. *a feasible point u^* is called* globally Pareto optimal *if there exists no feasible point $u \in \mathcal{U}$ dominating u^*. The image $J(u^*)$ of a globally Pareto optimal point u^* is called a* globally Pareto optimal value*. If this property holds in a neighborhood $U(u^*) \subset \mathcal{U}$, then u^* is called* locally Pareto optimal*.*

3. *the set of non-dominated feasible points is called the* Pareto set $\mathcal{P}_S$*, its image the* Pareto front $\mathcal{P}_F$.

A consequence of Definition 2 is that for each point that is contained in the Pareto set (the red line in Figure 1 (a)), one can only improve one objective by accepting a trade-off in at least one other objective. Figuratively speaking, in a two-dimensional problem, we are interested in finding the "lower left" boundary of the feasible set in objective space (cf. Figure 1 (b)).

(a)   (b)
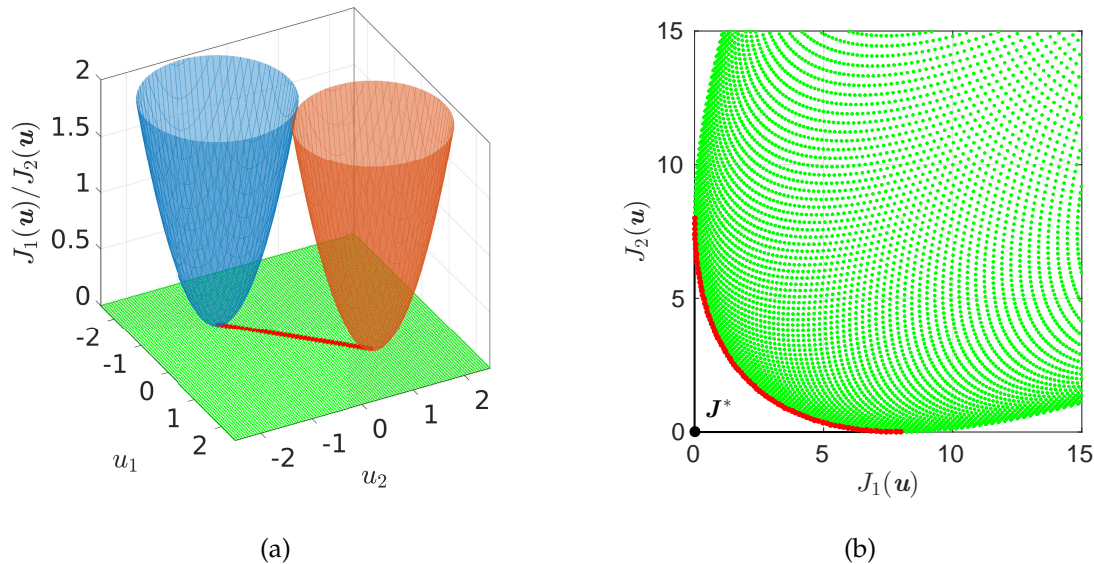
Figure 1. The red lines are the Pareto set (a) and Pareto front (b) of an exemplary multiobjective optimization problem (two paraboloids) of the form $\min_{u \in \mathbb{R}} J(u)$, $J : \mathbb{R}^2 \to \mathbb{R}^2$. The point $J^* = (0,0)^\top$ is called the utopian point.

Similar to single objective optimization, a necessary condition for optimality is based on the gradients of the objective functions. The first order conditions were independently discovered by Karush in 1939 [13] and by Kuhn and Tucker in 1951 [14]. Due to this, they are widely known as the *Karush-Kuhn-Tucker (KKT)* conditions:

Theorem 1 ([14])**.** *Let u^* be a Pareto optimal point of problem* (MOP) *and assume that $\nabla h_j(u^*)$ for $j = 1, \ldots, m$, and $\nabla g_s(u^*)$ for $s = 1, \ldots, l$, are linearly independent. Then there exist non-negative scalars $\alpha_1, \ldots, \alpha_k \geq 0$ with $\sum_{i=1}^{k} \alpha_i = 1$, $\gamma \in \mathbb{R}^m$ and $\mu \in \mathbb{R}^l$ such that*

$$\sum_{i=1}^{k} \alpha_i \nabla J_i(u^*) + \sum_{j=1}^{m} \gamma_j \nabla h_j(u^*) + \sum_{s=1}^{l} \mu_s \nabla g_s(u^*) = 0,$$

$$
\begin{aligned}
h_j(u^*) &= 0, & j &= 1, \ldots, m, \\
g_s(u^*) &\leq 0, & s &= 1, \ldots, l, \\
\mu_s g_s(u^*) &= 0, & s &= 1, \ldots, l, \\
\mu_s &\geq 0, & s &= 1, \ldots, l.
\end{aligned}
$$

(KKT)

The set of points satisfying these conditions is called the *set of substationary points* $\mathcal{P}_{S,\text{sub}}$ [4]. Obviously, $\mathcal{P}_{S,\text{sub}}$ is a superset of the Pareto set $\mathcal{P}_S$. Many algorithms for MOPs compute the set of substationary points, in particular gradient-based methods as we will see in the next section. This set can be reduced to the Pareto set in a consecutive step by performing a (comparatively cheap) non-dominance test.

2.2. Solution methods

Many researchers in multiobjective optimization focus their attention on developing efficient algorithms for the computation of Pareto sets. Algorithms for solving MOPs can be compiled into several fundamentally different categories of approaches. The first category is based on *scalarization techniques*, where ideas from single objective optimization theory are extended to the multiobjective situation. All scalarization techniques have in common that the Pareto set is approximated by a finite set of Pareto optimal points which are computed by solving scalar subproblems. Consequently, the resulting solution methods involve solving multiple optimization problems consecutively. Scalarization can be achieved by various approaches such as the weighted-sum method, the ϵ-constraint method, normal boundary intersection, or reference point methods [4,5,15].

Continuation methods make use of the fact that under certain conditions, the Pareto set is a smooth manifold of dimension $k-1$ [16]. This means that one can compute the tangent space in each point of the set, and a predictor step is performed in this space. The resulting point then has to be corrected to a Pareto optimal solution using a descend method [17].

Another prominent approach is based on *evolutionary algorithms* [18,19], where the underlying idea is to evolve an entire population of solutions during the optimization process. Significant advances have been made concerning multiobjective evolutionary algorithms (*MOEAs*) in recent years [20,21] (see also [22] for a survey) such that they are nowadays the most popular choice for solving MOPs due to the applicability to very complex problems and being easy to use in a black box fashion. Since convergence rates can be relatively slow for MOEAs, they can be coupled with locally fast methods close to the Pareto set. These approaches are known as *memetic algorithms*, see, e.g., [23–26].

Set-oriented methods provide an alternative deterministic approach to the solution of MOPs. Utilizing subdivision techniques, the desired Pareto set is approximated by a nested sequence of increasingly refined box coverings [27–29]. This way, a superset is computed which converges to the desired solution, even in situations where the Pareto set is disconnected. However, their complexity depends on both the dimension of the Pareto set as well as the decision space dimension. Due to this, one has to take additional steps to apply these algorithms for multiobjective optimal control problems.

Depending on the method of choice, gradient information can be used to accelerate convergence. While this is widely accepted in scalar-valued optimization, this is less the case when multiple objectives are present [30]. Nonetheless, many approaches exist where gradients are exploited, for example in order to create sequences converging to single points [31–33], to compute the entire set of valid decent directions [30,34], to obtain superlinear or quadratic convergence [35,36], or in combination with evolutionary approaches (memetic algorithms) [37–41]. In many of the gradient-based methods, the descent direction for all objectives is a convex combination of the individual gradients:

$$q(u) = \sum_{i=1}^{k} \hat{\alpha}_i \nabla J_i(u). \tag{1}$$

Here, $\hat{\alpha}$ is a fixed weight vector which is determined in such a way that

$$\langle q(u), \nabla J_i(u) \rangle < 0 \quad \forall\, i = 1, \ldots, k, \tag{2}$$

see, e.g., [31,32]. In the unconstrained case, there exists no u satisfying (2) only if $\|q(u)\| = 0$ which implies that u is substationary, cf. (KKT).

3. Surrogate models

The ever increasing computational capabilities allow us to analyze more and more complicated systems with a very large number of degrees of freedom, also in the context of optimal control, where practical problems range from process control [42,43] and energy management [15] over space mission design [44] to mobility and autonomous driving [45–48].

The above-mentioned examples can all be described by ordinary differential equations with a finite-dimensional state space $\mathcal{Y}$. In contrast to these problems, many phenomena in physics such as mechanical strain, heat flow, electromagnetism, fluid flow or even multi-physics simulations are governed by partial differential equations. Using numerical discretization schemes for the approximation of the spatial domain (such as finite elements or finite volumes) results in a very large number of degrees of freedom and a heavy computational burden. For more complex systems (such as turbulent flows [49]), simulating the dynamics is already very costly. Consequently, optimal control of these systems is all the more challenging, and considering multiple objectives further increases the cost. Due to this reason, only few problems have been addressed directly, see, e.g., [50–53]. A method exploiting special structure in the system dynamics has been proposed in [54], and a special case of Pareto optimal solutions – namely *Nash equilibria* – have been computed in [55,56][1].

A very popular approach to circumvent the problem of prohibitively large computational cost is the use of surrogate models. Here, the exact objective function $J(u)$ is replaced by a surrogate $J^r(u)$, where the superscript r stands for reduced. In many situations, this surrogate function can be evaluated faster by several orders of magnitude. The challenge here is to find a good trade-off between acceleration and model accuracy, and many approaches have been proposed over the past two decades.

In optimal control, there are two fundamentally different ways for model reduction. The first case – which is equivalently applicable to multiobjective optimization problems – is to directly derive a surrogate for the objective function, i.e., $J^r : \mathcal{U} \to \mathbb{R}^k$ is constructed by polynomials, radial basis functions or other means. In optimal control, an alternative way of reducing the computational effort is by introducing a reduced model for the system dynamics:

$$J^r(u) = \hat{J}^r(u, y) = \hat{J}(u, \mathcal{S}^r u),$$

where the reduced control-to-state operator $\mathcal{S}^r$ indicates that the model reduction is due to a surrogate model for the system dynamics.

In both situations, we cannot expect that $J^r(u) = J(u)$ holds for all $u \in \mathcal{U}$. Instead, we introduce an error which has to be taken into account. This is closely related to questions concerning uncertainty and noise. In this context, many researchers have addressed inaccuracies. In [58], the notion of ϵ-efficiency (cf. Definition 3) was first introduced in order to handle uncertain objective values. Uncertainty has also been considered in the context of multiobjective evolutionary computation, see, e.g., [59–61]. Alternative methods such as probabilistic [62,63], deterministic [64] or set-oriented approaches [65, 66] have also been proposed. The special case of many-objective optimization is covered in [67], applications are addressed in [68,69]. A different approach to uncertainties is via robust algorithms. Several examples from multiobjective optimization as well as optimal control can be found in [43,70–72].

In the following, we will first introduce some results concerning inaccuracies in multiobjective optimization in Section 3.1 and then give an overview of existing methods for both of the above-mentioned approaches, i.e., surrogate models for the objective function (Section 3.2) or for the system dynamics (Section 3.3). Since the first approach has already been covered extensively in several surveys [1–3], we only give a brief overview over the existing methods and the corresponding references.

3.1. Inaccuracies and ϵ-dominance

When using surrogate models in order to accelerate the solution process, we have to accept an error both in the objective function as well as the respective gradients. Furthermore, inaccuracies may occur due to stochastic processes or due to unknown model parameters. In these situations, the objective function and the corresponding gradients are only known approximately which has to be taken into account.

Suppose now that we only have approximations $J^r(u), \nabla J^r(u)$ of the objectives $J_i(u)$ and the gradients $\nabla J_i(u), i = 1, \ldots, k$, respectively. Furthermore, let us assume that upper bounds $\epsilon, \kappa \in \mathbb{R}^k$ for these errors are known:

$$\|J_i^r(u) - J_i(u)\|_2 \leq \epsilon_i \quad \forall u \in \mathcal{U}, \tag{3}$$

$$\|\nabla J_i^r(u) - \nabla J_i(u)\|_2 \leq \kappa_i \quad \forall u \in \mathcal{U}. \tag{4}$$

In this situation, we need to replace the dominance property 2 by an inexact version, also known as ϵ-dominance (see also [58,69]):

Definition 3 ([66]). *Consider the multiobjective optimization problem* (MOP) *where the objective function* $J(u)$ *is only known approximately according to* (3). *Then*

1. *a point u^* confidently dominates a point u, if $J^r(u^*) + \epsilon \leq J^r(u) - \epsilon$ and $J_i^r(u^*) + \epsilon_i < J_i^r(u) - \epsilon_i$ for at least one $i \in 1, \ldots, k$.*

---

[1]   When not using a priori selection methods such as scalarization or Nash equilibria, a decision maker to select the appropriate solution. This is called *Multicriteria Decision Making (MCDM)* [57] and is an entire area of research of its own. Thus, we will not go into further details about the decision making process here.

(a)   (b)   (c)   (d)   (e)
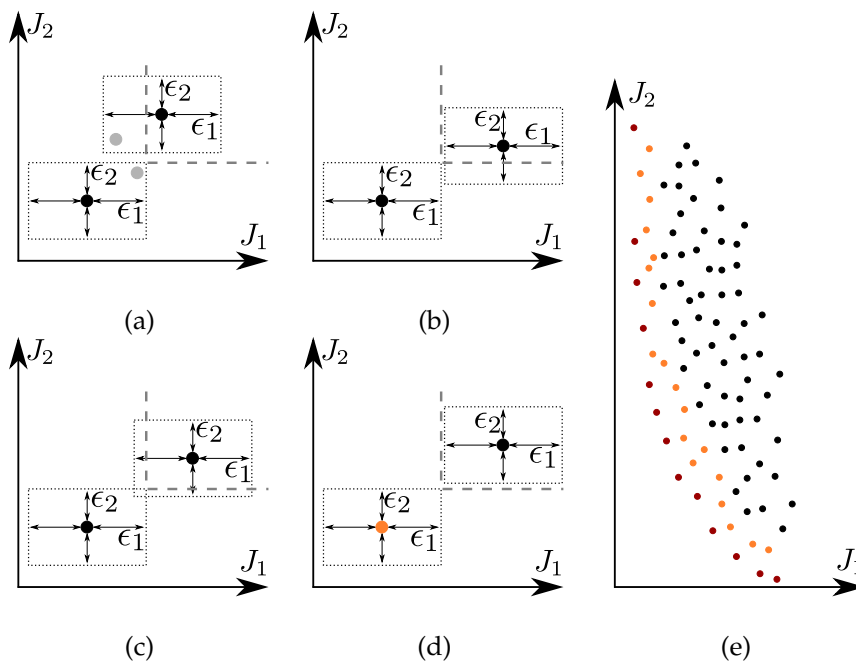
Figure 2. Example for the ϵ-dominance property. A point-wise comparison is illustrated in (a) – (d). The uncertainties are marked by the dashed boxes. Only in case (d) the lower left point confidently dominates the other point. In (e) the entire Pareto fronts for the exact problem ($\mathcal{P}_F$) and the inexact problem ($\mathcal{P}_{F,\epsilon}$) are shown in red and orange, respectively.

2. *The* set of almost non-dominated points *which is a superset of the Pareto set $\mathcal{P}_S$ is defined as:*

$$\mathcal{P}_{S,\epsilon} = \left\{ u^* \in \mathcal{U} \,\middle|\, \nexists u \in \mathcal{U} \text{ which confidently dominates } u^* \right\}. \tag{5}$$

The concept of ϵ-dominance is visualized in Figure 2. Theoretically, the true point could be anywhere inside the box defined by ϵ such that in the cases (a) to (c), the lower left point does not confidently dominate the other point. The necessary condition is violated for one component in (a) and (b), respectively, and for both components in (c). The gray points in 2 (a) show a possible realization of the true points in which no point is dominated by the other. In (d) the orange point confidently dominates the black one and in (e) we see the implications for the computation of Pareto fronts. Due to the inexactness, the number of points that are not confidently dominated is larger than in the exact case. This is also evident in Figure 3, where the exact and the inexact solution of an example problem from production [32] have been computed with an extension of the subdivision technique presented in [27], cf. [66] for details. Here, inexactness is introduced due to uncertainties in pricing. ϵ-dominance can be used for the development of algorithms for MOPs with uncertainties [59,61–63,65,67,68], for accelerating expensive MOPs [60,66,73] as well as for increasing the number of compromise solutions for the decision maker [63–65,69].

When considering gradient-based methods, inaccuracies in the gradients (Equation (4)) have to be taken into account. Since only approximations of the true gradient are known, these result in an inexact descent direction $q^r(u)$. The inaccuracy in the gradients introduces an upper bound for the angle between the individual gradients $\nabla J_i^r(u)$ and the descent direction $q^r(u)$. This is equivalent to a lower bound $\hat{\alpha}_{\min}$ for the weight vector, i.e., $\hat{\alpha}_i \in [0,1]$ in Equation (1) has to be replaced by $\hat{\alpha}_i^r \in [\hat{\alpha}_{\min,i}, 1]$ for $i = 1, \ldots, k$:

$$q^r(u) = \sum_{i=1}^{k} \hat{\alpha}_i^r \nabla J_i^r(u), \tag{6}$$

see [66] for a detailed discussion. The additional constraint ensures that if $q^r(u)$ is a descent direction for the inexact problem, it is also a descent direction for the true problem. Furthermore, we obtain a criterion for the accuracy up to which we can compute $\mathcal{P}_{S,\text{sub}}$ based on inexact gradient information:

(a)   (b)
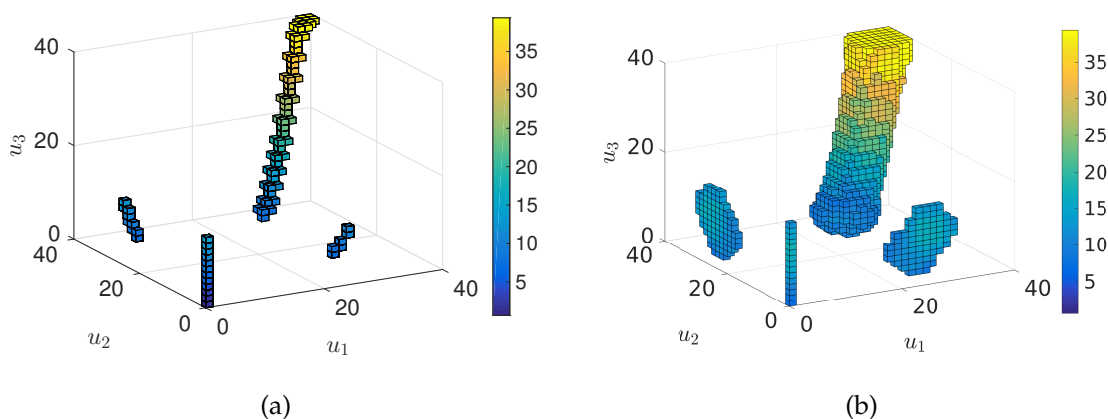
Figure 3. Example for an MOP from production [32] where inexactness is introduced due to uncertainties in pricing. The Pareto set $\mathcal{P}_S$ for the exact problem is shown in (a), the inexact set $\mathcal{P}_{S,\epsilon}$ is shown in (b).

Theorem 2 ([66]). *Consider the multiobjective optimization problem (MOP) without constraints. Only approximate gradients according to (4) are available and consequently, the descent direction is also only known approximately according to Equation (6). Assume that $\|q^r(u)\|_2 \neq 0$ and $\|\nabla J_i^r(u)\|_2 \neq 0$ for $i = 1, \ldots, k$. Let*

$$\hat{\alpha}_{\min,i} = \frac{1}{\|\nabla J_i^r(u)\|_2^2} \left(\|q^r(u)\|_2 \kappa_i - \sum_{\substack{j=1 \\ j \neq i}}^k \hat{\alpha}_j \left(\nabla J_i^r(u) \cdot \nabla J_i^r(u) \right) \right), \quad i = 1, \ldots, k.$$

Then the following statements are valid:

(a) *If $\sum_{i=1}^k \hat{\alpha}_{\min,i} > 1$ then there exists no direction $q(u)$ with*

$$\langle q(u), \nabla J_i(u) \rangle < 0 \quad \forall\, i = 1, \ldots, k,$$

  i.e. no descent direction for the exact problem.

(b) *All points u with $\sum_{i=1}^k \hat{\alpha}_{\min,i} = 1$ are contained in the set*

$$\mathcal{P}_{S,\kappa} = \left\{ u \in \mathbb{R}^n \,\middle|\, \left\| \sum_{i=1}^k \hat{\alpha}_i \nabla J_i(u) \right\|_2 \leq 2 \|\kappa\|_\infty \right\}. \tag{7}$$

A combination of Theorem 2 with the subdivision algorithm from [27] is shown in Figure 4. The algorithm constructs a nested sequence of increasingly refined box coverings which converges to the set of substationary points where in the unconstrained case, $\|q(u^*)\| = 0$ holds for all $u^* \in \mathcal{P}_{S,\text{sub}}$. The set $\mathcal{P}_{S,\text{sub}}$ is shown in red in Figure 4 (a). Due to the inexactness, we can no longer guarantee $\|q(u^*)\| = 0$. Instead, we obtain the set $\mathcal{P}_{S,\kappa}$ which is shown in green. The background is colored according to the optimality condition $\|q(u)\|$ of the exact problem and the dashed white line indicates the error bound (7) from the above theorem.

(a)   (b)
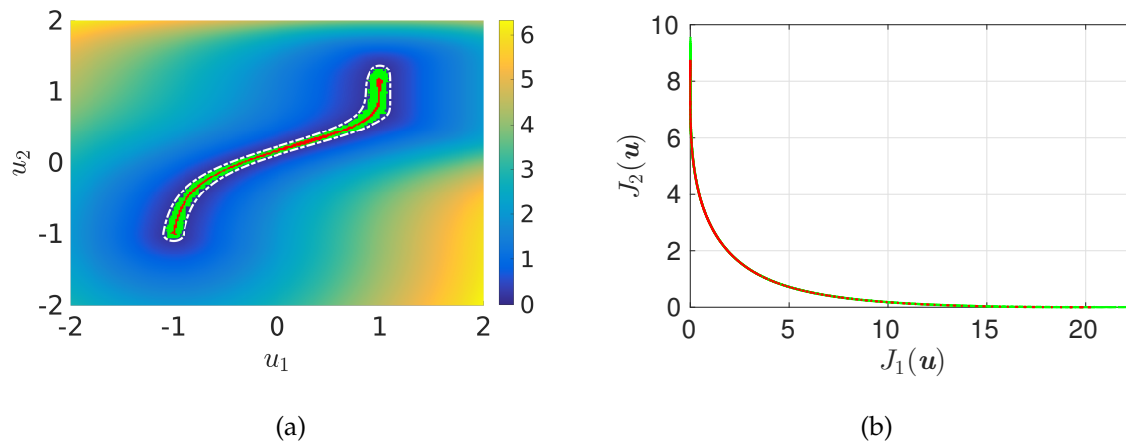
Figure 4. Exact and inexact solution ($\mathcal{P}_S$ and $\mathcal{P}_{S,\kappa}$) for a simple example with $J : \mathbb{R}^2 \to \mathbb{R}^2$, cf. [66] for details. (a) The sets $\mathcal{P}_S$ and $\mathcal{P}_{S,\kappa}$ (for a random error with $\kappa = (0.01, 0.01)^\top$) are shown in red and green, respectively. The background is colored according to the optimality condition $\|q(u)\|$ which has to be 0 for all substationary points. The dashed white line shows the error bound as derived in Theorem 2. (b) The corresponding Pareto fronts.

3.2. Surrogate models for the objective function

The most straight-forward approach for introducing a surrogate model is to directly construct a map from the control to the decision space. This means that only the essential input-output behavior

is covered whereas internal states as well as the system dynamics are neglected. Using such a *meta model*, one can very quickly obtain the objective function value for every u. This approach is equally applicable in finite-dimensional multiobjective optimization and has been used extensively in this context. To this end, we will only briefly cover the main questions that have to be addressed when using such models.

In special cases, one can exploit some structure in the problem formulation which yields a simplified analytic expression. However, in most cases, even if the equations can be written down in closed form, this approach requires a deep understanding of the underlying system which is often hard to get by. Moreover, small changes in the problem setup may result in having to repeat this tedious process all over. Due to these reasons, data-based approaches are used much more frequently. They are often easy to apply and much more general. In these approaches, the original objective function (or even a real-world experiment) is evaluated for a small number of inputs which are contained in the set $\mathcal{U}_{\text{ref}} = \{u^1, \ldots, u^p\}$, where p is the number of function evaluations (or *experiments*). The data points $J(u^1)$ to $J(u^p)$ are then used to fit the meta model such as, e.g., the coefficients of a polynomial basis. Obviously, the choice of suitable ansatz functions is essential for the success of a meta modeling strategy. Popular choices are

- response surface models (RSM),
- radial basis functions (RBF),
- statistical models such as Kriging or Gauss regression,
- machine learning methods such as Artificial Neural Networks or Support Vector Machines,

see [1] for an extensive survey in the context of multiobjective optimization. Additional surveys can be found in [74], where different statistical methods are compared, in [3,75] in relation to MOEAs, or in [76], where RSM and RBF are compared for crashworthiness problems.

Besides selecting the correct meta model, questions concerning the training data set have to be answered:

1. how large does the set $\mathcal{U}_{\text{ref}}$ have to be?
2. how can we pick the correct elements for $\mathcal{U}_{\text{ref}}$?
3. do we define $\mathcal{U}_{\text{ref}}$ in advance or online during the model building process?

In addition to the meta model, point 1 significantly depends on the problem under consideration. The more non-linear a problem is, the more data points are generally required to accurately construct a meta model. Obviously, the number also depends on point 2, the locations u^1 to u^k of these evaluations. The question of choosing the correct location is closely related to the field of *optimum experimental design* or *design of experiments (DoE)*, see, e.g., [77,78]. The relevant question there is how to optimally pick a set of experiments such that the overall approximation error becomes minimal. Such approaches have successfully been coupled with multiobjective optimization in [79,80]. Point 3 depends both on the meta modeling approach as well as the problem under consideration. In some situations (e.g., for real-world experiments), it may not be possible to iteratively determine the experiments. Instead, a *batch* approach has to be used. In computer experiments, flexibility is often higher such that an interplay between model building and high-fidelity evaluations can help to further reduce the number of experiments. In the context of machine learning, this process is also known as *active learning* [81,82].

Many algorithms combining multiobjective optimization and meta modeling have been proposed and there exists a vast literature concerning this topic. Table 1 on page 17 mentions both survey articles and a (non-exhaustive) selection of popular methods.

3.3. Surrogate models for the dynamical system

The above-mentioned meta modeling methods are widely studied and have been successfully applied in a large variety of multiobjective optimization problems. In the context of control, there exists an alternative option which is to derive a surrogate model for the system dynamics, i.e., to replace the high-fidelity control-to-state operator $\mathcal{S}$ by a cheaper surrogate model $\mathcal{S}^r$. Since the largest part

of the computational effort is due to solving the dynamical system, by this the reduced cost function $J^r(u) = \hat{J}^r(u, y) = \hat{J}(u, \mathcal{S}^r u)$ is also much cheaper to evaluate.

The use of *reduced order models (ROMs)* is not limited to control but has been used successfully in a large variety of multi-query problems [12], and several extensive surveys have been written on reduced order modeling and using ROMs in prediction, uncertainty quantification or optimization, see [83–85]. For nonlinear problems, the two most widely-used approaches are the *reduced basis method* and *Proper Orthogonal Decomposition (POD)* (also known as *Principal Component Analysis (PCA)* or *Karhunen-Loève Decomposition*).

3.3.1. ROMS via Proper Orthogonal Decomposition or the reduced basis method

Numerically solving a PDE is generally realized by discretizing the spatial domain with a numerical mesh using finite differences, finite elements or finite volumes. By this, the infinite-dimensional state space $\mathcal{Y}$ is transformed into a finite-dimensional space $\mathcal{Y}^N$ via Galerkin projection:

$$y(x, t) \approx y^N(x, t) = \sum_{i=1}^{N} z_i^N(t) \phi_i(x). \tag{8}$$

Here, N denotes the number of degrees of freedom and $\phi_i(x)$ are basis functions with local support such as indicator functions or hat functions. This transforms the PDE into an N-dimensional ordinary differential equation for the coefficients z. For complex domains as well as complex dynamics, the dimension can easily reach the order of millions such that solving the problem in $\mathcal{Y}^N$ can quickly become very expensive which is particularly challenging in the multi-query case.

The general concept in projection-based model reduction is therefore to find an appropriate space $\mathcal{Y}^r$ with dimension $\ell \ll N$ in which the system dynamics can nevertheless be approximated with sufficient accuracy. The two most common approaches to do this are the *reduced basis (RB) method* and *Proper Orthogonal Decomposition (POD)*. In both cases, we compute s so-called *snapshots* of the high-dimensional system and then use the data set $\{y_1^N, \dots, y_s^N\}$ to construct a reduced basis $\psi = \{\psi_1, \dots, \psi_\ell\}$ such that

$$y(x, t) \approx y^N(x, t) \approx y^r(x, t) = \sum_{i=1}^{\ell} z_i^\ell(t) \psi_i(x).$$

The following example describes this approach in more detail. For an extensive introduction, the reader is referred to [86,87].

Example 1 (Heat equation). *Suppose we want to solve the time-dependent heat equation on a domain Ω with homogeneous Neumann conditions on the boundary Σ:*

$$\begin{aligned} y_t(x, t) - \lambda \Delta y(x, t) &= 0, & (x, t) &\in \Omega \times (t_0, t_e], \\ y(x, 0) &= y_0(x), & x &\in \Omega, \\ y_n(x, t) &= 0, & (x, t) &\in \Sigma \times (t_0, t_e]. \end{aligned} \tag{9}$$

Here, y is the temperature and λ is the heat conductivity. The subscripts t and n indicate the derivatives with respect to time and the outward normal vector of the boundary, respectively. We now derive the weak form of (9) by multiplying with a test function φ and integrating over the domain Ω. Using Gauss's theorem, we obtain the following equation:

$$\int_\Omega y_t(\cdot, t)\varphi - \lambda \nabla y(\cdot, t) \cdot \nabla \varphi \, dx = 0, \qquad (x, t) \in \Omega \times (t_0, t_e]. \tag{10}$$

If we want to solve (10) *using the finite element method, we have to insert the Galerkin ansatz* (8) *into* (10) *and individually take each of the basis functions as a test function. By this, we obtain the following system of equations:*

$$\int_{\Omega} \sum_{i=1}^{N} z_{i,t}^{N}(t)\phi_i\phi_j - \lambda z_i^{N}(t)\nabla\phi_i \cdot \nabla\phi_j \, dx = 0, \qquad j = 1,\ldots,N. \tag{11}$$

Introducing the mass matrix $M \in \mathbb{R}^{N \times N}$ *and the* stiffness matrix $K \in \mathbb{R}^{N \times N}$ *with*

$$M_{i,j} = \int_{\Omega} \phi_i\phi_j \, dx, \qquad K_{i,j} = \int_{\Omega} \nabla\phi_i \cdot \nabla\phi_j \, dx,$$

this yields the following N-dimensional linear system:

$$Mz_t^{N}(t) - \lambda K z^{N}(t) = 0.$$

If we now want to compute a reduced order model instead of a high-dimensional finite element approximation, we can apply the same procedure except that now we have to use the reduced basis in the Galerkin ansatz as well as for test functions.

The most important difference between RB and POD is the area of application although this is not a strict separation. RB is mostly applied to parameter-dependent yet time-independent (i.e. elliptic) problems whereas POD (introduced in [88]) is applied to time-dependent problems described by parabolic or hyperbolic PDEs. Consequently, in RB the snapshots $\{y_1^N, \ldots, y_s^N\}$ are solutions corresponding to parameters $\{u_1, \ldots, u_s\}$ and in POD they are snapshots in time, collected at the time instants $\{t_0, \ldots, t_{s-1}\}$. Using an equidistant time grid h, the snapshots are taken at $\{t_0, t_0 + h \ldots, t_0 + (s-1)h\}$. In RB, the snapshots $\{y_1^N, \ldots, y_s^N\}$ often directly serve as the basis ψ. For time-dependent problems, this can cause numerical difficulties since some snapshots might be very similar (e.g., for very slow systems or periodic dynamics) such that the snapshot matrix $S = (y_1^N, \ldots, y_s^N)$ is ill-conditioned. Due to this, a *singular value decomposition* is performed on S and the ℓ leading left singular vectors are taken as the basis ψ. This results in an orthonormal basis which can be shown to be optimal with respect to the L^2 projection error [87,88]. Furthermore, the truncation error is given by the sum over the neglected singular values:

$$\epsilon_\ell = \frac{\sum_{i=\ell+1}^{s} \sigma_i}{\sum_{j=1}^{s} \sigma_j}.$$

Whereas the error between the infinite-dimensional solution and the solution via a standard discretization approach can be neglected in many situations, the error of the ROM depends on several factors such as the reference data, the basis size and the parameter or control for which the ROM is evaluated. Consequently, this error can be significant such that proper care has to be taken. The most common approach is to derive bounds either for the error of the reduced state $\|y^r - y^N\|$ or – in the case of optimal control – of the optimal solution $\|u^r - u^N\|$ obtained using the ROM, see, e.g., [89–93] for POD or [94–97] for RB methods. In addition, there are other measures that can be taken such as deriving balanced input-output behavior [98,99] or introducing additional terms [100] or modifications [101] in the POD-based ROM. For more detailed introductions to RB and POD, the reader is referred to [97] and [87,88], respectively.

3.3.2. Optimal control using surrogate models

There is a rich literature on optimal control of PDEs using surrogate models. The approaches can be summarized in three main categories:

1. build a model once,
2. construction of regular updates in a *trust region* approach,

3. construction of regular updates using error estimators.

Whereas the first category is the most efficient one (see, e.g., [102] for optimal control of the Navier–Stokes equations), it is in general not possible to prove convergence of the resulting algorithm.

In the second approach (which was developed by Fahl [103] for POD-based ROMs and one objective), one defines a trust region within which the current surrogate model is considered as trustworthy, see Figure 5 for an illustration. The ROM-based optimal control problem is then solved with the additional constraint that the solution has to remain within the trust region, i.e. $\|u^i - u^{\text{ref}}\| \leq \delta^i$, where i is the current step of the iterative optimization scheme and δ^i is the current trust region radius. After having obtained u^i, the high-dimensional system is evaluated and the improvement of the full system is determined:

$$\rho = \frac{|J^N(u^i) - J^N(u^{i-1})|}{|J^r(u^i) - J^r(u^{i-1})|}.$$

If ρ is close to 1, then the ROM is sufficiently accurate and the iterate u^i is accepted. We then use the high-dimensional solution to construct the next ROM at $u^{\text{ref}} = u^i$. If, on the other hand, ρ is close to 0, then the ROM accuracy was bad and the iterate u^i is rejected. Instead, the trust region radius δ^i is reduced and the optimal control problem is solved again. Using the trust region (TR-POD) approach, one can ensure convergence to the optimal solution of the high-dimensional problem. In case of the Navier–Stokes equations, this has been shown for different problem setups [103,104].
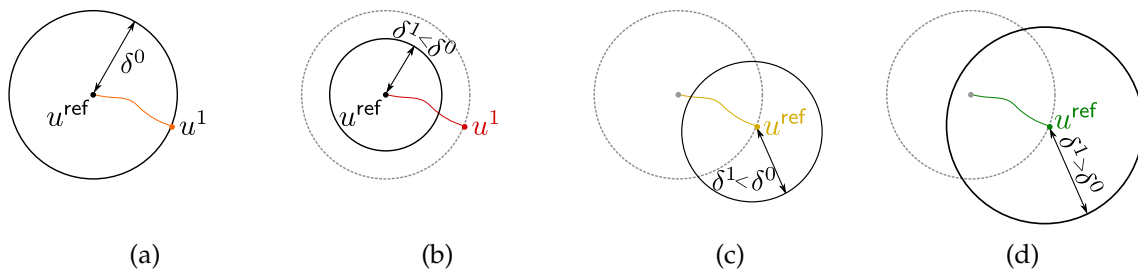
Figure 5. Trust region method. (a) The ROM-based optimal control problem is solved within the trust region δ^0. (b) If the improvement is poor for the full system (i.e., ρ is small), then the trust region radius is reduced and we repeat the computation with the same problem. (c) If the improvement is acceptable (intermediate values of ρ), then we compute a new model and proceed with a smaller trust region $\delta^1 < \delta^0$. (d) If the improvement is good (i.e., $\rho \approx 1$), then the trust region radius is increased.

In the third approach, error estimators Δ^J for the current iterate u^i are required. By evaluating $\Delta^J(u^i)$ it is possible to efficiently estimate the error between the high- and the low-dimensional solution, since

$$|J^r(u^i) - J^N(u^i)| < \Delta^J.$$

If this error estimate is larger than some prescribed upper bound ϵ, then the ROM has to be updated using data from a high-dimensional solution. Detailed information on error estimates can be found in, e.g., [91–97].

3.4. ROM-based multiobjective optimal control of PDEs

All three approaches for using ROMs in optimal control have recently been extended to multiobjective optimal control problems. Besides different ROM techniques, different algorithms for MOPs have been used as well such that a variety of methods has evolved, each of which is well-suited for certain situations.
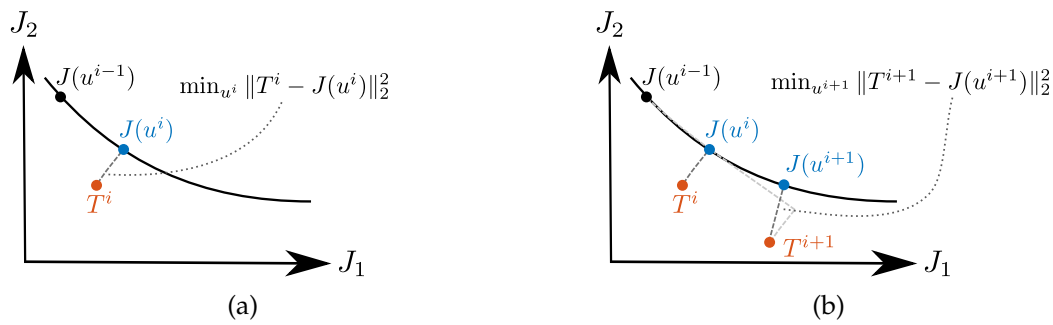
Figure 6. Reference point method. (a) Determination of a Pareto optimal solution by solving (13). (b) Determination of the consecutive point on the Pareto front by adjusting the target and solving the next scalar problem.

3.4.1. Scalarization

A natural and widely used approach to MOPs is via scalarization. By this, the vector of objectives is synthesized into a scalar objective function and the MOP is transformed into a sequence of scalar optimization problems for different scalarization parameters. In terms of ROM-based optimal control, this is advantageous because many techniques from scalar-valued optimal control can be extended. The main difference is now that the objective function may have a more complicated structure. In [105,106], the *weighted sum method* has been used in combination with RB in order to solve MOPs constrained by elliptic PDEs. In the weighted sum method, scalarization is achieved via convex combination of the individual objectives using the weight vector α:

$$\min_{u \in \mathcal{U}} \overline{J}(u) = \min_{u \in \mathcal{U}} \sum_{i=1}^{k} \alpha_i J_i(u). \tag{12}$$

The weighted sum method is probably the most straight-forward approach for including ROMs in MOPs. However, the method has strong limitations in the situation of non-convex problems, where it is impossible to compute the entire Pareto set [5].

A more advanced approach is the so-called *reference point method* [5] (cf. Figure 6 for an illustration), where the distance to an infeasible *target point T* with $T < J(u)$ has to be minimized:

$$\min_{u \in \mathcal{U}} \overline{J}(u) = \min_{u \in \mathcal{U}} \|T - J_i(u)\|. \tag{13}$$

By adjusting the target, we can move along the Pareto front and hence obtain an approximately equidistant covering of the front. The reference point method has been coupled with all three of the above-mentioned ROM approaches. In [107], it was used for multiobjective optimal control of the Navier–Stokes equations using one reduced model (cf. Figure 7 (a)). Here, the objectives are to stabilize a periodic solution – the well-known *von Kármán vortex street* – and to minimize the control cost at the same time. In [73] the trust region framework by Fahl (TR-POD, [103]) was extended (cf. Figure 7 (b)–(c) for a heat flow problem with a tracking and a cost minimization objective). The third ROM approach was used in [108,109]. The difficulty here is that the minimization of the distance to the target point results in a more complicated objective function which has to be treated carefully.

Scalarization techniques generally have the same limitations on the decision space dimension as scalar valued optimal control problems. This means that very efficient techniques (both direct and indirect) exist for high-dimensional controls. However, the number of objectives is limited because the parametrization in the scalarization step becomes extremely tedious and it is almost impossible to obtain a good approximation of the entire Pareto set for more than three objectives.
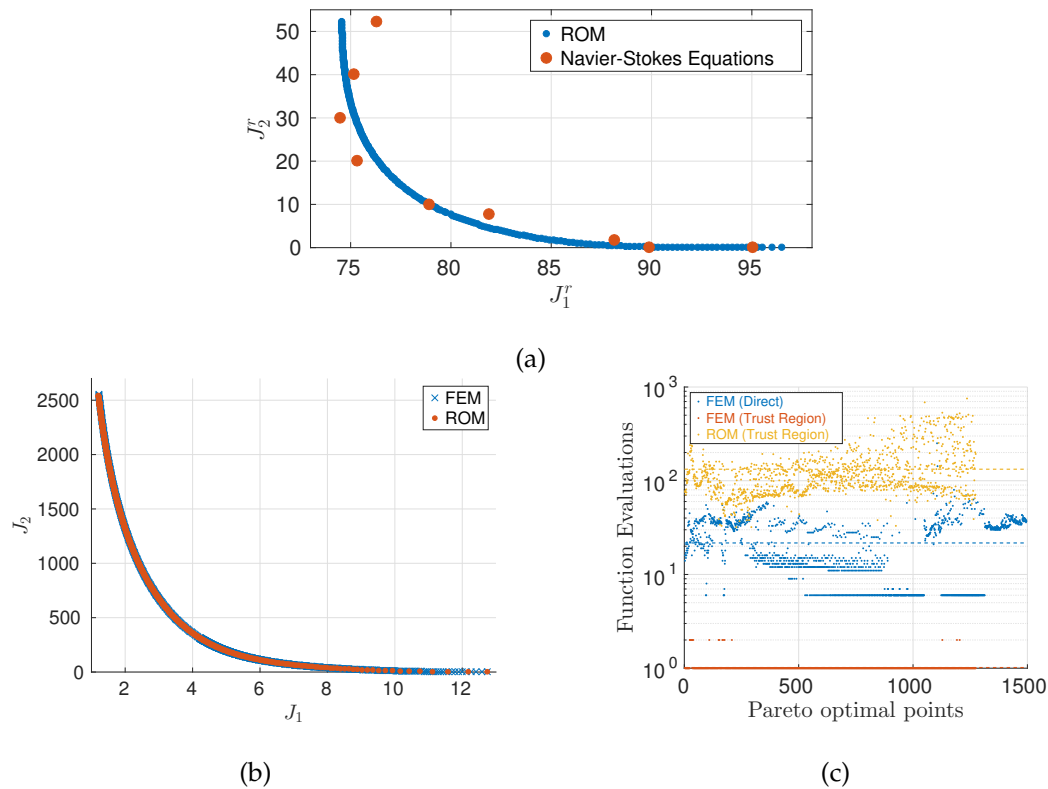
Figure 7. (a) Pareto front for an MOCP involving the Navier–Stokes equations (flow stabilization vs. cost), solved by coupling of a ROM (created once in advance) with the reference point method. Although we observe acceptable agreement, convergence cannot be guaranteed. (b) Pareto front for a heat flow MOCP (reference tracking vs. cost), solved by the TR-POD approach coupled with the reference point method. Convergence is achieved while reducing the number of expensive finite element (FEM) evaluations by a factor of ≈ 22, cf. (c).

3.4.2. Set-oriented approaches with ϵ-dominance

In contrast to scalarization, in set-oriented techniques the Pareto set is approximated by a box covering [27–29]. Here the limitations are contrary, meaning that the dimension of the decision space is rather limited while the number of objectives does not pose any problems for the algorithms. In practice however, the computational cost increases exponentially with the number of objectives (i.e. the dimension of the Pareto set) such that we are still limited to a moderate number of objectives.

First results coupling the subdivision algorithm developed in [27] with error estimates for POD-based ROMs have recently appeared [110,111]. In the subdivision algorithm, the decision space is divided into boxes which are alternatingly *subdivided* and *selected*. In the subdivision step, each existing box is subdivided into two smaller boxes. In the selection step, all boxes are eliminated which are dominated, i.e., they do not cover any part of the Pareto set. Numerically, this is realized by representing a box by a finite number of sample points and then marking a box as dominated if all sample points are dominated by samples from another box in the covering, see [27] for details.

The subdivision algorithm can be extended to inaccuracies by replacing the strict dominance test by an ϵ-dominance test as presented in Section 3.1 (see also Figure 3). After fixing an upper bound ϵ, we have to ensure that the surrogate models we use do not violate this bound anywhere in the control domain. Since this is cannot be achieved with a single ROM, one has to use multiple, locally valid ROMS instead (cf. Figure 8 (c)). The covering by local ROMs is managed in such a way that all points in the neighborhood around the reference u^{ref} at which the data for the ROM was collected satisfy the prescribed error bound ϵ. This way, the number of solutions of the high-dimensional system can
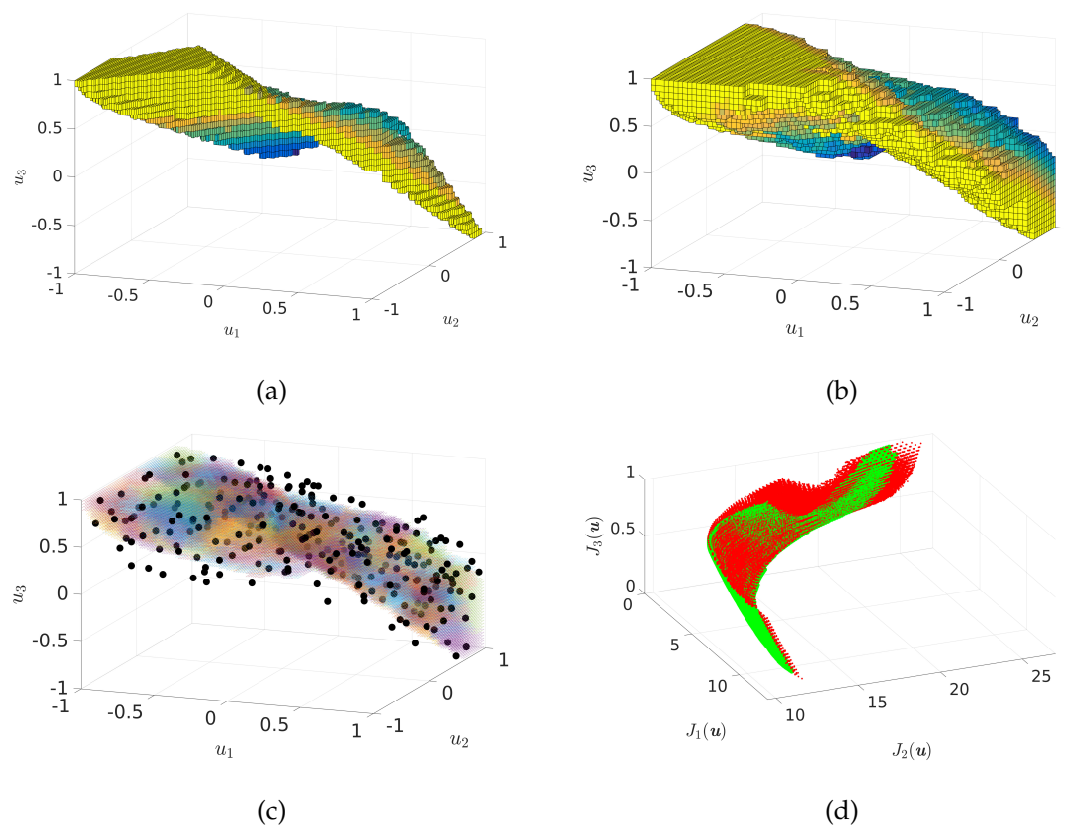
Figure 8. (a) Pareto set of a semilinear heat flow MOP with four controls (coloring according to u_4), solved directly including an FEM model in the subdivision algorithm. (b) Pareto set of the same problem, solved with localized ROMs. (c) The reference controls for which the local ROMs have been computed are shown in black, the colored dots are sample points at which the objective function was evaluated. The colorings denote assignments to a specific ROM. (d) The corresponding Pareto fronts, where the FEM solution is shown in green and the ROM solution in red.

be reduced significantly. A comparison between the exact and the ROM-based solution is shown in Figure 8 for a semilinear heat flow MOCP with two tracking type objectives and a cost minimization objective. Due to the ROM approach, the number of evaluations of the FEM model could be reduced by a factor of ≈ 1000.

3.5. Summary

Before moving on to feedback control, a summary of the relevant publications where multiobjective optimization and meta modeling interact is given in Table 1. The references are categorized into surveys, algorithms using surrogate models for the objective function, the system dynamics, and specific reduction approaches for MOPs. Furthermore, some applications are referenced.

4. Feedback control

Even when the objective function is not very expensive to evaluate, MOCPs often have a large computational cost, see, e.g., [123] for various examples. This becomes a limiting factor in situations where the solution time is critical as is the case in real-time applications. Due to the increasing computational power as well as the advances in algorithms, *model predictive control (MPC)* (see [124,125] for extensive introductions) has become a very powerful and widely-used method for realizing model-based feedback control of complex systems.

Table 1.   Overview of publications (in chronological order) where surrogate modeling and multiobjective optimization are combined.

Surveys	
Tabatabei et al. [2], Chugh et al. [3]	Extensive surveys on meta modeling for MOEAs
Voutchkov & Keane [74], Knowles & Nakayama [1], Jin [75]	Surveys on meta modeling approaches from statistics (RSM, RBF) and machine learning in combination with MOEAs
Benner et al. [83], Taira et al. [85], Peherstorfer et al. [84]	Surveys on reduced order modeling of dynamical systems

Algorithms using meta models for the objective function	
Ong et al. [112], Ray et al. [113]	Combination of RBF and MOEA
Chung & Alonso [114], Keane [115]	Combination of Kriging models and MOEA
Karakasis & Giannakoglou [116]	RBF as a cheap pre-processing step in a MOEA
Knowles [117]	Combination of DoE and an interactive method
Zhang et al. [118]	Combination of Gaussian process models and scalarization
Telen et al. [79]	Combination of DoE and scalarization and MOEA
Chugh et al. [119]	Kriging model in combination with reference vector approach for MaOPs

Meta models specifically tailored to multiobjective optimization	
Shimoyama et al. [120]	Kriging surrogate for hypervolume approximation (MOEA)
Pan et al. [121]	Surrogate model for dominance relations with uncertainties

Algorithms using surrogate models for the system dynamics	
Iapichino et al. [105]	Combination of POD and weighted sum
Banholzer et al. [108,109]	Combination of POD and reference point method
Iapichino et al. [106]	Combination of RB and weighted sum
Peitz [73]	Combination of TR-POD and reference point method
Beermann et al. [110,111]	Combination of POD and set-oriented method

Applications	
Albunni et al. [53]	POD and MOEA applied to Maxwell equation
Ma & Qu [80]	MO of a switched reluctance motor by coupling RSM and MOEA (particle swarm optimization)
Peitz et al. [107]	POD-based multiobjective optimal control of the Navier–Stokes equations via scalarization and set-oriented methods
Wang et al. [122]	MOEA with multifidelity surrogate-management and offline-online decomposition applied to a Trauma system

In MPC, an optimal control problem is solved on a short time horizon (the *prediction horizon*) while the real system (the *plant*) is running. Then the first entry of the optimal control is applied to the plant and the process is repeated with the time frame moving forward by one *sample time h*, see Figure 9 for an illustration. This way, a closed-loop behavior is achieved. On the downside, we have to solve the optimal control problem within the sample time h. This can be in the order of seconds or minutes (in the case of chemical processes) down to a few microseconds, for example in power electronics applications.
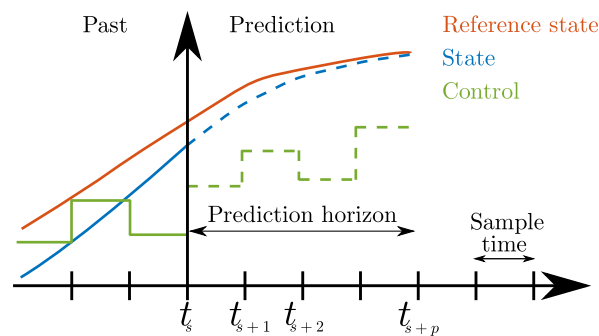
Figure 9. Sketch of the MPC method. Due to the real-time constraints, the optimization problem has to be solved faster than the sample time h.

In many MPC problems, stabilization of the system with respect to some reference state is the most important aspect. Nevertheless, there exist variations where stability is not an issue such that other (more economic) objectives can be pursued. These methods are known as *economic MPC* [126,127]. Another variation is the so-called *explicit MPC* [128], where the optimal control is computed in advance for a large number of different states and stored in a library. This way, the computational effort is shifted to an offline phase and during operation, we only have to select the optimal control from the library.

As in multiobjective optimal control, there are numerous applications where multiple objectives are interesting in feedback control. This means that we would have to solve the problem (MOCP) with $t_0 = t_s$ and $t_e = t_{s+p}$ repeatedly within the sample time h. It is immediately clear that even the simplest MOCPs cannot be solved fast enough to allow for real-time applicability. Consequently, efficient algorithms have to be developed which can be divided into approaches where one Pareto optimal solution is computed online and approaches with an offline phase during which the MOCP is solved[2].

When implementing a MOMPC algorithm, one should keep in mind that – regardless of the algorithm used – the resulting trajectory need not be Pareto optimal, even if each single step is, cf. [129]. A remedy to this issue is presented in [130], where the selection of compromise solutions is restricted to a part of the Pareto front which is determined in the first MPC step. Due to this, upper bounds for the objective function values can be guaranteed.

4.1. Online multiobjective optimization

In the classical MPC framework, the optimal control problem is solved online within the sample time h. Since it is in general impossible to approximate the entire Pareto set sufficiently accurately within this time frame, there are three alternatives:

1. compute a single Pareto optimal solution according to some predefined preference,
2. compute only a rough approximation of the Pareto set,
3. compute an arbitrary Pareto optimal control which satisfies additional constraints (e.g., stability).

In the first approach, the objective function is scalarized using, for instance, the weighted sum method (12) or the reference point method (13). In this situation, well-established approaches from scalar-valued MPC exist which one can build on. First results using the weighted sum method have appeared in [131]. In [132,133], the authors use the same scalarization for an MPC problem with convex objective functions. In this situation, it is guaranteed that any Pareto optimal solution can

---

[2]   In this article, algorithms where Artificial Neural Networks (ANN) have to be trained beforehand are nonetheless assigned to the first category if the optimization is performed entirely online.

be computed using weighted sums, and the weights can be adapted online according to a decision maker's preference. Due to the convexity, stability can be proved for the resulting MPC algorithm. In [134], this approach is extended by providing gradient information of the objectives with respect to the weight vector α. This way, the weights can be adapted in such a way that a desired change in the objective space is realized. For non-convex problems, the weighted sum method is incapable of computing the entire Pareto set. Therefore, in [135], a variation of the reference point method is applied, where the target T is the utopian point J^*, i.e. the vector of the individual minima. This way, also non-convex problems can be treated. In fact, due to the reference point method, the objective function is always convex [109] which can be exploited during the optimization. Alternative scalarization methods are the *ϵ-constraint method* [136] or *lexocographic ordering* [137].

A disadvantage of a priori scalarization is that it is often difficult to select the scalarization parameter in such a way that a desired trade-off solution is obtained, and the remedy proposed in [134] is only applicable to a specific class of problems. Therefore, an alternative approach is to quickly compute a rough approximation of the entire Pareto set and then select the desired control online. Such methods have been proposed by many authors. The general approach is to us a MOEA and stop the computation after few iterations. In the next step, one of these suboptimal solutions is selected. This selection is realized by specifying a weight vector for the objectives in [138–141] and by the satisficing trade-off method in [142].

As a third option, we can compute a single Pareto optimal point without specifying which one we are specifically interested in as long as it satisfies additional constraints such as the stability of the system. Approaches of this type have been developed in [136] and [143], where a game theoretic approach is used.

4.2. Offline-Online decomposition

A well-known trick to avoid heavy online computations is to introduce an offline-online decomposition (very similar to meta modeling approaches where surrogate models are constructed before solving the MOCP). This means that the Pareto set is computed beforehand and in the online phase, an optimal compromise is selected according to a decision maker's preference or some heuristic based on the system state or the environment.

Many of the approaches which fall in this category use a standard feedback controller instead of MPC, see [144] for a short review concerning methods using scalarization and offline PID controller optimization. In the offline phase, a Pareto set is computed for the controller parameters. Possible objectives are – among many others – overshooting behavior, energy efficiency or robustness. Algorithms of this type have been proposed in [145–147] using MOEAs and in [148–150] using set oriented methods.

An alternative approach is motivated by explicit MPC, i.e., the idea to solve many MOPs offline such that the correct solution can be extracted from a library in the online phase. Such a method has been proposed in [48]. In the offline phase, one has to identify all possible scenarios which can occur in the online phase. Such a scenario consists of both system states as well as constraints. This results in a large number of MOPs which have to be solved. In order to reduce this number, symmetries in the problem are exploited. To this end, a concept known as *motion primitives* [151,152] is extended. In short, this means that if

$$\arg\min_{u \in \mathcal{U}} \text{MOP}_1 = \arg\min_{u \in \mathcal{U}} \text{MOP}_2,$$

where MOP_1 and MOP_2 are two problem instances from the offline library, then we only have to solve one of the problems in order to have a Pareto optimal solution for both. Moreover, if two problem instances vary only slightly, one can use a previously computed solution as a good initial guess for the next MOCP to further decrease the computational effort [73]. In the online phase, the correct Pareto set is selected from the library (according to the system state and the constraints) and an optimal compromise is selected according so a decision maker's preference α. In contrast to the affine linear
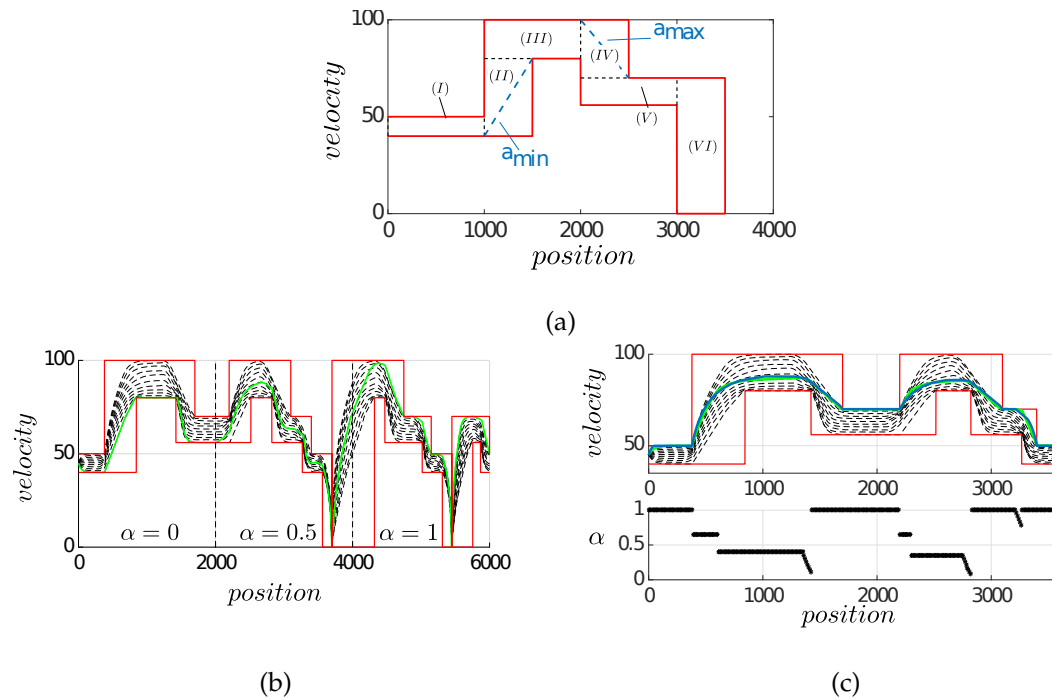
Figure 10. Results for the offline-online MPC approach from [48]. (a) Different constraint scenarios ((I) to (VI)), i.e., constant velocity, acceleration, deceleration, stopping. (b) Example track driven with the MPC algorithm. The red lines define the velocity bounds, the black dashed lines are trajectories corresponding to a constant weight α, and the green line is a trajectory where the weight is changed from 0 (energy efficient) over 0.5 (average) to 1 (fast). (c) Comparison between the MPC algorithm (coupled with a simple heuristic for the weighting) and the global optimum obtained via dynamic programming.

solutions which can be computed in explicit MPC for linear-quadratic problems, one has to rely on interpolation between solutions in the nonlinear setting.

4.2.1. Example: Autonomous driving

We here want to demonstrate the superiority of multiobjective approaches over scalar-valued MPC using the example of autonomously driving electric vehicles [47,48,153]. The problem there is to find the set of optimal engine torque profiles such that the velocity is maximized while the energy consumption is minimized. Additional constraints have to be taken into account such as speed limits or stop signs. The system dynamics are described by a four-dimensional, highly nonlinear ODE for the vehicle velocity, the battery state of charge and two battery voltage drops, cf. [153] for details. Numerical investigations reveal several symmetries in the system such that the only relevant state for a scenario is the current velocity whereas all other states only have a minor influence on the solution of the MOP. Consequently, the velocity as well as the constraints form the above-mentioned scenarios, see Figure 10 (a) for an illustration. For example, a scenario could be that the current velocity is 60 km/h and that the speed limit is currently increasing from 50 to 100 km/h, cf. scenario (II) in Figure 10 (a). We then solve the MOP for this scenario and store the Pareto set in a library. By discretizing the velocity into steps of 0.1 (i.e., $v(t_0) = \ldots, 59.9, 60.0, 60.1, \ldots$), we have to solve 1727 MOPs in total in the offline phase.

In the online phase, we now select the relevant Pareto set from the library and – according to a decision maker's preference – apply one of the Pareto optimal controls to the electric vehicle. This is done repeatedly such that a feedback loop is realized. The result is illustrated in Figure 10 (b) for an example track, where the black lines correspond to constant weighting of the two criteria and the

Table 2. Overview of publications (in chronological order) for multiobjective feedback control.

Algorithms without offline phase - computation of single points

Kerrigan et al. [131], Wojsznis et al. [154]	Scalarization via weighted sum (WS)
Kerrigan & Maciejowski [155], He et al. [137]	Scalarization via lexicographic ordering
Bemporad & Muñoz de la Peña [132,133]	Scalarization via WS for convex objectives, guaranteed stability for large gain vs. noise robust stabilizing objectives
Geisler & Trächtler [134]	WS, online adaptation of weights using gradient information
Maestre et al. [143]	Scalarization via game-theoretic approach
Zavala & Flores-Tlacuahuac [135]	Scalarization via reference point approach
Hackl et al. [129]	Scalarization via WS for linear time-invariant (LTI) systems
Zavala [136]	Scalarization via ϵ-constraint: economic objective, stability as constraint
Grüne & Stieler [130]	Economic objectives, performance bounds via selection criterion

Algorithms without offline phase - approximation of entire Pareto set

Laabidi et al. [138,140], Garcìa et al. [141]	Artificial neural network (ANN) for state prediction, optimization via MOEA, selection of Pareto point via WS
Bouani et al. [139]	ANN for state prediction, comparison of two MOEAs and WS for MOP
Nakayama et al. [142]	Few MOEA iterations online, selection via satisficing trade-off method

Algorithms with offline phase

Fonseca [145], Herreros et al. [146]	Offline computation of Pareto optimal controller parameters using MOEA
Scherer et al. [156]	Robust control using a common Lyapunov function for multiple stability criteria
Ben Aicha et al. [147]	Offline computation of Pareto optimal controllers parameters via EA and WS, online selection according to geometric mean of objectives
Krüger et al. [148]	Offline computation of Pareto optimal controllers parameters via Set oriented methods, parametric model reduction for increased efficiency
Hernández et al. [149] Xiong et al. [150]	Offline computation of Pareto optimal controllers parameters via simple cell mapping
Peitz et al. [48]	Offline-online decomposition similar to explicit MPC

Applications

Zambrano & Camacho [157]	MOMPC of a solar refrigeration plant via scalarization
Porfírio et al. [158]	MOMPC of an industrial splitter using a min-max reformulation
Pedersen & Yang [159]	MO PID controller design for magnetic levitation systems via MOEA
Li et al. [160]	Multiobjective adaptive cruise control for vehicles
Hu et al. [161]	MOMPC of high-power converters via WS
Núñez et al. [162]	MOMPC of dynamic pickup and delivery problems using MOEA
Peitz et al. [163]	MOMPC of an industrial laundry, scalarization of a traveling salesman problem via WS

green line corresponds to a varying weight. This way, a flexible cruise control is established where the driver can quickly adapt, for instance, to changing energy requirements.

As has been mentioned before, an alternative to interactively choosing a weight is to implement some heuristic which automatically chooses a weight based on the current situation. Such an approach is visualized in Figure 10 (c), where the weighting depends on the vehicle velocity as well as on current and future velocity constraints. For a simpler track it is possible to compute a globally optimal solution for a scalarized objective using dynamic programming. We see that with the heuristic, the MOMPC approach yields trajectories close to the global optimum while only having finite horizon information.

4.3. Summary

We again conclude the section by giving a summary of publications where multiobjective optimization is applied in a real-time context. The publications are divided into four categories. The first two contain algorithms where the MOP is solved online. The categories differ in whether a single point is computed or the entire set is approximated. Consequently an offline phase is not required except in the case where surrogate models are trained in order to accelerate the online computations. The third category then contains the methods with an offline optimization phase, and some applications are mentioned in the fourth category.

5. Reduction techniques for many-objective optimization problems

Another important restricting factor in multiobjective optimization is the number of objectives [164]. For MOPs with four or more objectives, the term *many-objective optimization (MaOP)* has been coined and over the past years, many researchers have dedicated their work to address MaOPs and the issues arising from the *curse of dimensionality*, cf. [165] for an overview, [166,167] for new concepts for identifying non-dominated solutions and [168–172] for evolutionary approaches.

A popular approach for MaOPs are interactive methods [173–178]. These methods do not compute the entire set of optimal compromises but instead interactively explore the Pareto set. Starting at the current Pareto optimal solution, a *decision maker* can choose in which direction to proceed, i.e., which objective to improve at the expense of some other, currently less important objective. The approach in [178], for example, allows for Pareto optimal movements both in decision and objective space. One of the main advantages of interactive methods, is the reduced computational effort – especially in the presence of many criteria – since it is not affected significantly by the dimension of the Pareto set. Moreover, this way decision making from a vast number of Pareto optimal solutions is avoided which can be overwhelming for a decision maker. Consequently, interpretability and usability are increased.

Besides interactive methods, several reduction techniques have been proposed in the context of many-objective optimization and although it is not the main theme of this review article, we want to give a brief overview of these reduction approaches since they also aim at increasing the efficiency of solving MOPs. These reduction techniques can be divided into two main categories. The first one is *objective reduction*, where the aim is to reduce the number of objectives while (approximately) preserving the Pareto set. The observation behind this is that not all objectives are of equal importance to the structure of the Pareto set, which is measured by the *degree of conflict* [179]. Consequently, when it is possible to identify the main contributors to the Pareto set, then one can solve a reduced problem taking into account only these most important objectives. Different approaches have been proposed for this identification step, all of which use a set of sample points. In [179], both exact and inexact algorithms are proposed for selecting a subset of objectives such that only those points in the Pareto set are lost which are worse in all remaining objectives by a constant δ or more. This approach is also exploited in [180]. In [181,182], POD (cf. Section 3.3) is used to identify such a subset, a related concept is implemented in [183] using hyperplanes. In [184], an entropy-based approach is presented and in [185], the relevant subset is selected multiple times within an evolutionary procedure. A slightly different approach is pursued in [186], where the authors split large decision spaces into several smaller ones according to the relevance of decision variables on specific objectives.

(a)   (b)   (c)
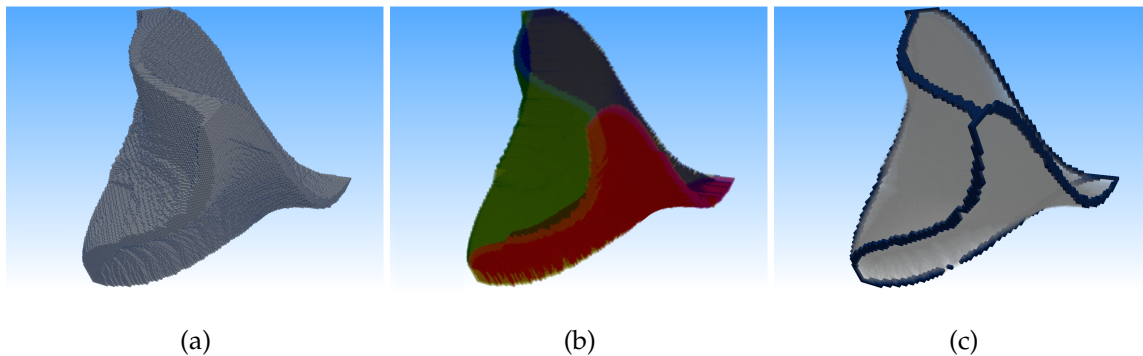
Figure 11. Visualization of the hierarchical structure of Pareto sets. (a) Pareto set of an example problem with $J : \mathbb{R}^3 \to \mathbb{R}^4$. (b) The four Pareto sets taking only three objectives into account form the boundary of the original Pareto set. (c) The Pareto sets in (b) are again bounded by the respective bi-objective subproblems.

The method proposed in [187] possesses characteristics of the first category – namely objective reduction – as well as of the second category which is the exploitation of the structure of Pareto sets and front. Therein, first the *corners* of the Pareto front are identified and this information is used to select the relevant objectives. Algorithms of the second type all exploit this hierarchical structure. This means that in under certain assumptions, the Pareto front is bounded by the Pareto fronts of subproblems where one or more objectives have been neglected, cf. [188,189]. This way, the solution can be computed by a hierarchical approach where – starting with scalar problems – the boundary is computed before finally the interior is obtained. Very recently, results about the hierarchical structure of the Pareto set, i.e., in decision space, have appeared, see [73,190] for details. This approach is illustrated in Figure 11 where the solution to an MOP with four objectives is shown as well as the Pareto sets of the subproblems with three and two objectives, respectively.

6. Future directions

This survey has given an overview over recent advances in the context of accelerating multiobjective optimization. These are surrogate models, feedback control and objective reduction techniques. Similar to almost every other field of science, it can be expected that the immense developments in data-based methods will also have a major impact on research in multiobjective optimization, in particular in the context of surrogate modeling. A very large number of researchers from the dynamical systems community is working on data-based methods using the *Koopman operator*, which is an infinite-dimensional but linear operator describing the dynamics of observables [191,192]. Significant effort has been put into the development of numerical methods for approximating this operator from data, see, e.g., [193,194]. This way, the dynamics of observations can be reconstructed entirely from data and without any knowledge of the underlying system dynamics. In a way, this allows is to merge the two surrogate modeling categories from Sections 3.2 and 3.3 since we can approximate the dynamics not only of the state but directly of the objectives. Several methods have recently been proposed to use the Koopman operator for data-based controller design, both in simulations [195–198] as well as experiments [199,200]. The results are very promising such that it is just a matter of time until these methods are utilized for multiobjective optimal control.

In the same manner, machine learning techniques [201] will very likely gain more and more attention, both in multiobjective optimization as well as optimal control. There are already many papers on this topic or related ones and the number is growing quickly.

Acknowledgement: This work is supported by the Priority Programme SPP 1962 "Non-smooth and Complementarity-based Distributed Parameter Systems" of the German Research Foundation (DFG).

Peer-reviewed version available at *Math. Comput. Appl.* **2018**, *23*, 30; doi:10.3390/mca23020030

Reference

1. Knowles, J.; Nakayama, H. Meta-Modeling in Multiobjective Optimization. In *Multiobjective Optimization: Interactive and Evolutionary Approaches*; Branke, J.; Deb, K.; Miettinen, K.; Slowinski, R., Eds.; Springer Berlin Heidelberg, 2008; pp. 245–284.

2. Tabatabaei, M.; Hakanen, J.; Hartikainen, M.; Miettinen, K.; Sindhya, K. A survey on handling computationally expensive multiobjective optimization problems using surrogates: non-nature inspired methods. *Structural and Multidisciplinary Optimization* **2015**, *52*, 1–25.

3. Chugh, T.; Sindhya, K.; Hakanen, J.; Miettinen, K. A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms. *Soft Computing* **2017**, pp. 1–30.

4. Miettinen, K. *Nonlinear Multiobjective Optimization*; Springer Science & Business Media, 2012.

5. Ehrgott, M. *Multicriteria optimization*, 2 ed.; Springer Berlin Heidelberg New York, 2005.

6. Hindi, H.A.; Hassibi, B.; Boyd, S.B. Multiobjective $\mathcal{H}_2/\mathcal{H}_\infty$-Optimal Control via Finite Dimensional Q-Parametrization and Linear Matrix Inequalities. Proceedings of the American Control Conference, 1998, pp. 3244–3249.

7. Zhu, Q.J. Hamiltonian Necessary Conditions for a Multiobjective Optimal Control Problem with Endpoint Constraints. *SIAM Journal on Control and Optimization* **2000**, *39*, 97–112.

8. Gambier, A.; Badreddin, E. Multi-objective optimal control: An overview. 16th IEEE International Conference on Control Applciations, 2007, pp. 170–175.

9. Tröltzsch, F. Optimal Control Of Partial Differential Equations. *Graduate studies in mathematics* **2010**, *112*.

10. Hinze, M.; Pinnau, R.; Ulbrich, M.; Ulbrich, S. *Optimization with PDE Constraints*; Springer Science+Business Media, 2009.

11. Ober-Blöbaum, S. Discrete Mechanics and Optimal Control. PhD thesis, University of Paderborn, 2008.

12. Schilders, W.H.A.; van der Vorst, H.A.; Rommes, J. *Model Order Reduction*; Springer Berlin Heidelberg, 2008.

13. Karush, W. Minima of Functions of Several Variables with Inequalities as Side Constraints. Master thesis, University of Chicago, Illinois, 1939.

14. Kuhn, H.W.; Tucker, A.W. Nonlinear programming. Proceedings of the 2nd Berkeley Symposium on Mathematical and Statsitical Probability. University of California Press, 1951, pp. 481–492.

15. Romaus, C.; Böcker, J.; Witting, K.; Seifried, A.; Znamenshchykov, O. Optimal energy management for a hybrid energy storage system combining batteries and double layer capacitors. Energy Conversion Congress and Exposition, 2009. ECCE 2009. IEEE. IEEE Xplore Digital Library, Los Alamitos, 2009, pp. 1640–1647.

16. Hillermeier, C. *Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach*; Birkhäuser, 2001.

17. Schütze, O.; Dell'Aere, A.; Dellnitz, M. On Continuation Methods for the Numerical Treatment of Multi-Objective Optimization Problems. Dagstuhl Seminar Proceedings, 2005.

18. Deb, K. *Multi-objective optimization using evolutionary algorithms*; Vol. 16, John Wiley & Sons, 2001.

19. Coello Coello, C.A.; Lamont, G.B.; Van Veldhuizen, D.A. *Evolutionary Algorithms for Solving Multi-Objective Problems*; Vol. 2, Springer Science & Business Media, 2007.

20. Zitzler, E.; Laumanns, M.; Thiele, L. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. *TIK-Report 103* **2001**.

21. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* **2002**, *6*, 182–197.

22. Zhou, A.; Qu, B.Y.; Li, H.; Zhao, S.Z.; Suganthan, P.N.; Zhang, Q. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation* **2011**, *1*, 32–49.

23. Ong, Y.S.; Lim, M.H.; Chen, X. Memetic Computation—Past, Present & Future. *IEEE Computational Intelligence Magazine* **2010**, *5*, 24–31.

24. Neri, F.; Cotta, C.; Moscato, P. *Handbook of memetic algorithms*; Vol. 379, Springer, 2012.

25. Schütze, O.; Alvarado, S.; Segura, C.; Landa, R. Gradient subspace approximation: a direct search method for memetic computing. *Soft Computing* **2016**.

26. Schütze, O.; Martín, A.; Lara, A.; Alvarado, S.; Salinas, E.; Coello Coello, C.A. The directed search method for multi-objective memetic algorithms. *Computational Optimization and Applications* **2016**, *63*, 305–332.

27.   Dellnitz, M.; Schütze, O.; Hestermeyer, T. Covering Pareto sets by Multilevel Subdivision Techniques. *Journal of Optimization Theory and Applications* **2005**, *124*, 113–136.

28.   Jahn, J. Multiobjective Search Algorithm with Subdivision Technique. *Computational Optimization and Applications* **2006**, *35*, 161–175.

29.   Schütze, O.; Witting, K.; Ober-Blöbaum, S.; Dellnitz, M. Set Oriented Methods for the Numerical Treatment of Multiobjective Optimization Problems. In *EVOLVE - A Bridge between Probability, Set Oriented Numerics and Evolutionary Computation*; Tantar, E.; Tantar, A.A.; Bouvry, P.; Del Moral, P.; Legrand, P.; Coello Coello, C.A.; Schütze, O., Eds.; Springer Berlin Heidelberg, 2013; Vol. 447, *Studies in Computational Intelligence*, pp. 187–219.

30.   Bosman, P.A.N. On Gradients and Hybrid Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation* **2012**, *16*, 51–69.

31.   Fliege, J.; Svaiter, B.F. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research* **2000**, *51*, 479–494.

32.   Schäffler, S.; Schultz, R.; Weinzierl, K. Stochastic Method for the Solution of Unconstrained Vector Optimization Problems. *Journal of Optimization Theory and Applications* **2002**, *114*, 209–222.

33.   Gebken, B.; Peitz, S.; Dellnitz, M. A Descent Method for Equality and Inequality Constrained Multiobjective Optimization Problems. *arXiv:1712.03005* **2017**.

34.   Bosman, P.A.N.; de Jong, E.D. Exploiting gradient information in numerical multi-objective evolutionary optimization. Proceedings of the 2005 conference on Genetic and evolutionary computation (GECCO 05), 2005, pp. 755–762.

35.   Fliege, J.; Grana Drummond, L.M.; Svaiter, B.F. Newton's method for multiobjective optimization. *SIAM Journal on Optimization* **2009**, *20*, 602–626.

36.   Fliege, J.; Vaz, A.I.F. A SQP type method for constrained multiobjective optimization. *Optimization Online* **2015**, pp. 1–35.

37.   Brown, M.; Smith, R.E. Directed multi-objective optimization. *International Journal of Computers, Systems, and Signals* **2005**, *6*, 3–17.

38.   Harada, K.; Sakuma, J.; Kobayashi, S. Local Search for Multiobjective Function Optimization: Pareto Descent Method. Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation (GECCO 06). ACM, 2006, pp. 659–666.

39.   Harada, K.; Sakuma, J.; Ono, I.; Kobayashi, S. Constraint-Handling Method for Multi-objective Function Optimization: Pareto Descent Repair Operator. In *Evolutionary Multi-Criterion Optimization*; Obayashi, S.; Deb, K.; Poloni, C.; Hiroyasu, T.; Murata, T., Eds.; Springer Berlin Heidelberg, 2007; pp. 156–170.

40.   Custódio, A.L.; Madeira, J.F.A.; Vaz, A.I.F.; Vicente, L.N. Direct multisearch for multiobjective optimization. *SIAM Journal on Optimization* **2011**, *21*, 1109–1140.

41.   Désidéri, J.A. Mutiple-Gradient Descent Algorithm for Multiobjective Optimization. European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS); Eberhardsteiner, J.; Böhm, H.J.; Rammerstorfer, F.G., Eds., 2012, pp. 3974–3993.

42.   Garduno-Ramirez, R.; Lee, K.Y. Multiobjective optimal power plant operation through coordinate control with pressure set point scheduling. *IEEE Transactions on Energy Conversion* **2001**, *16*, 115–122.

43.   Logist, F.; Houska, B.; Diehl, M.; van Impe, J. Robust multi-objective optimal control of uncertain (bio)chemical processes. *Chemical Engineering Science* **2011**, *66*, 4670–4682.

44.   Ober-Blöbaum, S.; Ringkamp, M.; zum Felde, G. Solving Multiobjective Optimal Control Problems in Space Mission Design using Discrete Mechanics and Reference Point Techniques. 51st IEEE International Conference on Decision and Control; , 2012; pp. 5711–5716.

45.   Lu, J.; DePoyster, M. Multiobjective optimal suspension control to achieve integrated ride and handling performance. *IEEE Transactions on Control Systems Technology* **2002**, *10*, 807–821.

46.   Geisler, J.; Witting, K.; Trächtler, A.; Dellnitz, M. Multiobjective optimization of control trajectories for the guidance of a rail-bound vehicle. IFAC Proceedings Volumes (IFAC-PapersOnline), 2008, Vol. 17, pp. 4380–4386.

47.   Dellnitz, M.; Eckstein, J.; Flaßkamp, K.; Friedel, P.; Horenkamp, C.; Köhler, U.; Ober-Blöbaum, S.; Peitz, S.; Tiemeyer, S. Multiobjective Optimal Control Methods for the Development of an Intelligent Cruise Control. Progress in Industrial Mathematics at ECMI 2014; Russo, G.; Capasso, V.; Nicosia, G.; Romano, V., Eds. Springer, 2017, pp. 633–641.

48. Peitz, S.; Schäfer, K.; Ober-Blöbaum, S.; Eckstein, J.; Köhler, U.; Dellnitz, M. A Multiobjective MPC Approach for Autonomously Driven Electric Vehicles. *IFAC PapersOnLine* **2017**, *50*, 8674–8679.

49. Brunton, S.L.; Noack, B.R. Closed-Loop Turbulence Control: Progress and Challenges. *Applied Mechanics Reviews* **2015**, *67*, 1–48.

50. Vemuri, V.R.; Cedeno, W. A New Genetic Algorithm for Multi-objective Optimization in Water Resource Management. *Proceedings of 1995 IEEE International Conference on Evolutionary Computation* **1995**, *1*, 1–11.

51. Rosehart, W.D.; Cañizares, C.A.; Quintana, V.H. Multiobjective optimal power flows to evaluate voltage security costs in power networks. *IEEE Transactions on Power Systems* **2003**, *18*, 578–587.

52. Lotov, A.V.; Kamenev, G.K.; Berezkin, V.E.; Miettinen, K. Optimal control of cooling process in continuous casting of steel using a visualization-based multi-criteria approach. *Applied Mathematical Modelling* **2005**, *29*, 653–672.

53. Albunni, M.N.; Rischmuller, V.; Fritzsche, T.; Lohmann, B. Multiobjective Optimization of the Design of Nonlinear Electromagnetic Systems Using Parametric Reduced Order Models. *IEEE Transactions on Magnetics* **2009**, *45*, 1474–1477.

54. Ober-Blöbaum, S.; Padberg-Gehle, K. Multiobjective optimal control of fluid mixing. Proceedings in Applied Mathematics and Mechanics (PAMM), 2015, pp. 639–640.

55. Ramos, A.M.; Glowinski, R.; Periaux, J. Nash Equilibria for the Multiobjective Control of Linear Partial Differential Equations 1. *Journal of Optimization Theory and Applications* **2002**, *112*, 457–498.

56. Borzì, A.; Kanzow, C. Formulation and Numerical Solution of Nash Equilibrium Multiobjective Elliptic Control Problems. *SIAM Journal on Control and Optimization* **2013**, *51*, 718–744.

57. Triantaphyllou, E. Multi-criteria decision making methods. In *Multi-criteria decision making methods: A comparative study*; Springer, 2000; pp. 5–21.

58. White, D.J. Epsilon efficiency. *Journal of Optimization Theory and Applications* **1986**, *49*, 319–337.

59. Hughes, E.J. Evolutionary Multi-objective Ranking with Uncertainty and Noise. In *Evolutionary Multi-Criterion Optimization: First International Conference, Zurich, Switzerland, March 7-9*; Zitzler, E.; Thiele, L.; Deb, K.; Coello Coello, C.A.; Corne, D., Eds.; Springer Berlin Heidelberg, 2001; pp. 329–343.

60. Deb, K.; Mohan, M.; Mishra, S. Evaluating the epsilon-Domination Based Multi-Objective Evolutionary Algorithm for a Quick Computation of Pareto-Optimal Solutions. *Evolutionary Computation* **2005**, *13*, 501–525.

61. Basseur, M.; Zitzler, E. A Preliminary Study on Handling Uncertainty in Indicator-Based Multiobjective Optimization. Applications of Evolutionary Computing: EvoWorkshops 2006: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoINTERACTION, EvoMUSART, and EvoSTOC, Budapest, Hungary; Rothlauf, F.; Branke, J.; Cagnoni, S.; Costa, E.; Cotta, C.; Drechsler, R.; Lutton, E.; Machado, P.; Moore, J.H.; Romero, J.; Smith, G.D.; Squillero, G.; Takagi, H., Eds., 2006, pp. 727–739.

62. Teich, J. Pareto-Front Exploration with Uncertain Objectives. In *Evolutionary Multi-Criterion Optimization*; Zitzler, E.; Thiele, L.; Deb, K.; Coello Coello, C.A.; Corne, D., Eds.; Springer Berlin Heidelberg, 2001; pp. 314–328.

63. Schütze, O.; Coello Coello, C.A.; Tantar, E.; Talbi, E.G. Computing the Set of Approximate Solutions of an MOP with Stochastic Search Algorithms. Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation. ACM, 2008, pp. 713–720.

64. Engau, A.; Wiecek, M.M. Generating ϵ-efficient solutions in multiobjective programming. *European Journal of Operational Research* **2007**, *177*, 1566–1579.

65. Hernández, C.; Sun, J.Q.; Schütze, O., Computing the Set of Approximate Solutions of a Multi-objective Optimization Problem by Means of Cell Mapping Techniques. In *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation IV: International Conference held at Leiden University, July 10-13, 2013*; Emmerich, M.; Deutz, A.; Schütze, O.; Bäck, T.; Tantar, E.; Tantar, A.A.; Moral, P.D.; Legrand, P.; Bouvry, P.; Coello, C.A., Eds.; Springer International Publishing, 2013; pp. 171–188.

66. Peitz, S.; Dellnitz, M. Gradient-based multiobjective optimization with uncertainties. In *NEO 2016*; Maldonado, Y.; Trujillo, L.; Schütze, O.; Riccardi, A.; Vasile, M., Eds.; Springer, 2018; Vol. 731, pp. 159–182.

67. Farina, M.; Amato, P. A fuzzy definition of "optimality" for many-criteria optimization problems. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* **2004**, *34*, 315–326.

68. Singh, A.; Minsker, B.S. Uncertainty-based multiobjective optimization of groundwater remediation design. *Water Resources Research* **2008**, *44*.

69. Schütze, O.; Vasile, M.; Coello Coello, C.A. Computing the Set of epsilon-efficient Solutions in Multi-Objective Space Mission Design. *Journal of Aerospace Computing, Information, and Communication* **2009**, *8*, 53–70.

70. Deb, K.; Gupta, H.V. Searching for Robust Pareto-optimal Solutions in Multi-objective Optimization. In *Evolutionary multi-criterion optimization*; Coello Coello, C.A.; Hernández Aguirre, A.; Zitzler, E., Eds.; Springer Berlin Heidelberg, 2005; pp. 150–164.

71. Xue, Y.; Li, D.; Shan, W.; Wang, C. Multi-objective Robust Optimization Using Probabilistic Indices. Third International Conference on Natural Computation, 2007, Vol. 4, pp. 466–470.

72. Dellnitz, M.; Witting, K. Computation of robust Pareto points. *International Journal of Computing Science and Mathematics* **2009**, *2*, 243–266.

73. Peitz, S. Exploiting Structure in Multiobjective Optimization and Optimal Control. PhD thesis, Paderborn University, 2017.

74. Voutchkov, I.; Keane, A.J. Multiobjective optimization using surrogates. Adaptive Computing in Design and Manufacture (ACDM), 2006, pp. 167–175.

75. Jin, Y. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation* **2011**, *1*, 61–70.

76. Fang, H.; Rais-Rohani, M.; Liu, Z.; Horstemeyer, M.F. A comparative study of metamodeling methods for multiobjective crashworthiness optimization. *Computers and Structures* **2005**, *83*, 2121–2136.

77. Sacks, J.; Welch, W.J.; Mitchell, T.J.; Wynn, H.P. Design and Analysis of Computer Experiments. *Statistical Science* **1989**, *4*, 409–435.

78. Atkinson, A.C.; Donev, A.N.; Tobias, R.D. *Optimum Experimental Designs, With SAS*; Oxford University Press, 2007.

79. Telen, D.; Logist, F.; Van Derlinden, E.; Tack, I.; Van Impe, J. Optimal experiment design for dynamic bioprocesses: A multi-objective approach. *Chemical Engineering Science* **2012**, *78*, 82–97.

80. Ma, C.; Qu, L. Multiobjective Optimization of Switched Reluctance Motors Based on Design of Experiments and Particle Swarm Optimization. *IEEE Transactions on Energy Conversion* **2015**, *30*, 1144–1153.

81. Cohn, D.A.; Ghahramani, Z.; Jordan, M.I. Active learning with statistical models. *Journal of Artificial Intelligence Research* **1996**.

82. Yu, K.; Bi, J.; Tresp, V. Active Learning via Transductive Experimental Design. Proceedings of the 23rd international conference on Machine learning. ACM, 2006, pp. 1081–1088.

83. Benner, P.; Gugercin, S.; Willcox, K. A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems. *SIAM Review* **2015**, *57*, 483–531.

84. Peherstorfer, B.; Willcox, K.; Gunzburger, M.D. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *ACDL Technical Report TR16-1* **2016**, pp. 1–57.

85. Taira, K.; Brunton, S.L.; Dawson, S.T.M.; Rowley, C.W.; Colonius, T.; McKeon, B.J.; Schmidt, O.T.; Gordeyev, S.; Theofilis, V.; Ukeiley, L.S. Modal Analysis of Fluid Flows: An Overview. *AIAA Journal* **2017**, *55*, 4013–4041.

86. Brenner, S.L.; Scott, R. *The Mathematical Theory of Finite Element Methods*, 3 ed.; Springer Science & Business Media, 2003.

87. Volkwein, S. Model reduction using proper orthogonal decomposition. *Lecture Notes* **2011**, pp. 1–43.

88. Sirovich, L. Turbulence and the dynamics of coherent structures part I: coherent structures. *Quarterly of Applied Mathematics* **1987**, *XLV*, 561–571.

89. Kunisch, K.; Volkwein, S. Control of the Burgers Equation by a Reduced-Order Approach Using Proper Orthogonal Decomposition. *Journal of Optimization Theory and Applications* **1999**, *102*, 345–371.

90. Kunisch, K.; Volkwein, S. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM Journal on Numerical Analysis* **2002**, *40*, 492–515.

91. Hinze, M.; Volkwein, S. Proper Orthogonal Decomposition Surrogate Models for Nonlinear Dynamical Systems: Error Estimates and Suboptimal Control. In *Reduction of Large-Scale Systems*; Benner, P.; Sorensen, D.C.; Mehrmann, V., Eds.; Springer Berlin Heidelberg, 2005; Vol. 45, pp. 261–306.

92. Tröltzsch, F.; Volkwein, S. POD a-posteriori error estimates for linear-quadratic optimal control problems. *Computational Optimization and Applications* **2009**, *44*, 83–115.

93. Lass, O.; Volkwein, S. Adaptive POD basis computation for parametrized nonlinear systems using optimal snapshot location. *Computational Optimization and Applications* **2014**, *58*, 645–677.

94.   Grepl, M.A.; Patera, A.T. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations.   _ESAIM: Mathematical Modelling and Numerical Analysis_ **2005**, _39_, 157–181.

95.   Veroy, K.; Patera, A.T. Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations: rigorous reduced-basis a posteriori error bounds. _International Journal for Numerical Methods in Fluids_ **2005**, _47_, 773–788.

96.   Haasdonk, B.; Ohlberger, M. Reduced Basis Method for Finite Volume Approximations of Parametrized Evolution Equations. _ESAIM: Mathematical Modelling and Numerical Analysis_ **2008**, _42_, 277–302.

97.   Rozza, G.; Huynh, D.B.P.; Patera, A.T. Reduced Basis Approximation and a Posteriori Error Estimation for Affinely Parametrized Elliptic Coercive Partial Differential Equations. _Archives of Computational Methods in Engineering_ **2008**, _15_, 229–275.

98.   Willcox, K.; Peraire, J. Balanced Model Reduction via the Proper Orthogonal Decomposition. _AIAA Journal_ **2002**, _40_, 2323–2330.

99.   Rowley, C.W. Model Reduction for Fluids, Using Balanced Proper Orthogonal Decomposition. _International Journal of Bifurcation and Chaos_ **2005**, _15_, 997–1013.

100.   Noack, B.R.; Papas, P.; Monkewitz, P.A. The need for a pressure-term representation in empirical Galerkin models of incompressible shear flows. _Journal of Fluid Mechanics_ **2005**, _523_, 339–365.

101.   Cordier, L.; Abou El Majd, B.; Favier, J. Calibration of POD reduced-order models using Tikhonov regularization. _International Journal for Numerical Methods in Fluids_ **2009**, _63_, 269–296.

102.   Bergmann, M.; Cordier, L.; Brancher, J.P. Optimal rotary control of the cylinder wake using proper orthogonal decomposition reduced-order model. _Physics of Fluids_ **2005**, _17_, 1–21.

103.   Fahl, M. Trust-region Methods for Flow Control based on Reduced Order Modelling. Phd thesis, University of Trier, 2000.

104.   Bergmann, M.; Cordier, L.; Brancher, J.P. Drag Minimization of the Cylinder Wake by Trust-Region Proper Orthogonal Decomposition. In _Active Flow Control_; King, R., Ed.; Springer, 2007; pp. 309–324.

105.   Iapichino, L.; Trenz, S.; Volkwein, S. Multiobjective optimal control of semilinear parabolic problems using POD. In _Numerical Mathematics and Advanced Applications (ENUMATH 2015)_; Karasözen, B.; Manguoglu, M.; Tezer-Sezgin, M.; Goktepe, S.; Ugur, Ö., Eds.; Springer, 2016; pp. 389–397.

106.   Iapichino, L.; Ulbrich, S.; Volkwein, S. Multiobjective PDE-Constrained Optimization Using the Reduced-Basis Method. _Advances in Computational Mathematics_ **2017**, _43_, 945–972.

107.   Peitz, S.; Ober-Blöbaum, S.; Dellnitz, M. Multiobjective Optimal Control Methods for Fluid Flow Using Model Order Reduction. _arXiv:1510.05819_ **2015**.

108.   Banholzer, S.; Beermann, D.; Volkwein, S. POD-Based Bicriterial Optimal Control by the Reference Point Method. _IFAC-PapersOnLine_ **2016**, _49_, 210–215.

109.   Banholzer, S.; Beermann, D.; Volkwein, S. POD-Based Error Control for Reduced-Order Bicriterial PDE-Constrained Optimization. _Annual Reviews in Control_ **2017**, _44_, 226–237.

110.   Beermann, D.; Dellnitz, M.; Peitz, S.; Volkwein, S. Set-Oriented Multiobjective Optimal Control of PDEs using Proper Orthogonal Decomposition. In _Reduced-Order Modeling (ROM) for Simulation and Optimization_; Springer, 2018; pp. 47–72.

111.   Beermann, D.; Dellnitz, M.; Peitz, S.; Volkwein, S. POD-based multiobjective optimal control of PDEs with non-smooth objectives. Proceedings in Applied Mathematics and Mechanics (PAMM), 2017, pp. 51–54.

112.   Ong, Y.S.; Nair, P.B.; Keane, A.J. Evolutionary Optimization of Computationally Expensive Problems via Surrogate Modeling. _AIAA Journal_ **2003**, _41_, 687–696.

113.   Ray, T.; Isaacs, A.; Smith, W. Surrogate Assisted Evolutionary Algorithm for Multi-Objective Optimization. In _Multi-Objective Optimization_; World Scientific, 2011; pp. 131–151.

114.   Chung, H.S.; Alonso, J. Mutiobjective Optimization Using Approximation Model-Based Genetic Algorithms. _10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference_ **2004**.

115.   Keane, A.J. Statistical Improvement Criteria for Use in Multiobjective Design Optimization. _AIAA Journal_ **2006**, _44_, 879–891.

116.   Karakasis, M.K.; Giannakoglou, K.C. Metamodel-Assisted Multi-Objective Evolutionary Optimization. EUROGEN 2005 - Evolutionary Methods for Design, Optimization and Control with Ap- plications to Industrial Problems; Schilling, R.; Haase, W.; Periaux, J.; Baier, H.; Bugeda, G., Eds., 2005.

117.   Knowles, J. ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation* **2006**, *10*, 50–66.

118.   Zhang, Q.; Liu, W.; Tsang, E.; Virginas, B. Expensive Multiobjective Optimization by MOEA / D with Gaussian Process Model. *IEEE Transactions on Evolutionary Computation* **2010**, *14*, 456–474.

119.   Chugh, T.; Jin, Y.; Miettinen, K.; Hakanen, J.; Sindhya, K. A Surrogate-Assisted Reference Vector Guided Evolutionary Algorithm for Computationally Expensive Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation* **2018**, *22*, 129–142.

120.   Shimoyama, K.; Jeong, S.; Obayashi, S. Kriging-Surrogate-Based Optimization Considering Expected Hypervolume Improvement in Non-Constrained Many-Objective Test Problems. 2013 IEEE Congress on Evolutionary Computation, 2013, pp. 658–665.

121.   Pan, L.; He, C.; Tian, Y.; Wang, H.; Zhang, X.; Jin, Y. A Classification Based Surrogate-Assisted Evolutionary Algorithm for Expensive Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation* **2018**.

122.   Wang, H.; Jin, Y.; Jansen, J.O. Data-Driven Surrogate-Assisted Multiobjective Evolutionary Optimization of a Trauma System. *IEEE Transactions on Evolutionary Computation* **2016**, *20*, 939–952.

123.   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* **2010**, *42*, 591–603.

124.   Allgöwer, F.; Zheng, A. *Nonlinear model predictive control*; Vol. 26, Birkhäuser, 2012.

125.   Grüne, L.; Pannek, J. *Nonlinear Model Predictive Control*, 2 ed.; Springer International Publishing, 2017.

126.   Rawlings, J.B.; Amrit, R. Optimizing process economic performance using model predictive control. In *Nonlinear model predictive control*; Springer Berlin Heidelberg, 2009; pp. 119–138.

127.   Grüne, L.; Müller, M.A.; Faulwasser, T. Economic Nonlinear Model Predictive Control. *Foundation and Trends in Systems and Control* **2018**, *5*, 1–98.

128.   Alessio, A.; Bemporad, A. A survey on explicit model predictive control. In *Nonlinear Model Predictive Control: Towards New Challenging Applications*; Magni, L.; Raimondo, D.M.; Allgöwer, F., Eds.; Springer Berlin Heidelberg, 2009; Vol. 384, pp. 345–369.

129.   Hackl, C..M.; Larcher, F.; Dötlinger, A.; Kennel, R.M. Is multiple-objective model-predictive control "optimal"? IEEE International Symposium on Sensorless Control for Electrical Drives and Predictive Control of Electrical Drives and Power Electronics (SLED/PRECEDE), 2013, pp. 1–8.

130.   Grüne, L.; Stieler, M. Performance Guarantees for Multiobjective Model Predictive Control. *Universität Bayreuth (https://epub.uni-bayreuth.de/3359/)* **2017**.

131.   Kerrigan, E.C.; Bemporad, A.; Mignone, D.; Morari, M.; Maciejowski, J.M. Multi-objective Prioritisation and Reconfiguration for the Control of Constrained Hybrid Systems. Proceedings of the American Control Conference, 2000, pp. 1694–1698.

132.   Bemporad, A.; Muñoz de la Peña, D. Multiobjective model predictive control. *Automatica* **2009**, *45*, 2823–2830.

133.   Bemporad, A.; Muñoz de la Peña, D. Multiobjective Model Predictive Control Based on Convex Piecewise Affine Costs. Proceedings of the European Control Conference, 2009, pp. 2402–2407.

134.   Geisler, J.; Trächtler, A. Control of the Pareto optimality of systems with unknown disturbances. IEEE International Conference on Control and Automation (ICCA), 2009, pp. 695–700.

135.   Zavala, V.M.; Flores-Tlacuahuac, A. Stability of multiobjective predictive control: A utopia-tracking approach. *Automatica* **2012**, *48*, 2627–2632.

136.   Zavala, V.M. A Multiobjective Optimization Perspective on the Stability of Economic MPC. *IFAC-PapersOnLine* **2015**, *48*, 974–980.

137.   He, D.; Wang, L.; Sun, J. On stability of multiobjective NMPC with objective prioritization. *Automatica* **2015**, *57*, 189–198.

138.   Laabidi, K.; Bouani, F. Genetic algorithms for multiobjective predictive control. First International Symposium on Control, Communications and Signal Processing, 2004, pp. 149–152.

139.   Bouani, F.; Laabidi, K.; Ksouri, M. Constrained Nonlinear Multi-objective Predictive Control. *IMACS Multiconference on "Computational Engineering in Systems Applications", CESA* **2006**, pp. 1558–1565.

140.   Laabidi, K.; Bouani, F.; Ksouri, M. Multi-criteria optimization in nonlinear predictive control. *Mathematics and Computers in Simulation* **2008**, *76*, 363–374.

141. García, J.J.V.; Garay, V.G.; Gordo, E.I.; Fano, F.A.; Sukia, M.L. Intelligent Multi-Objective Nonlinear Model Predictive Control (iMO-NMPC): Towards the 'on-line' optimization of highly complex control problems. *Expert Systems with Applications* **2012**, *39*, 6527–6540.

142. Nakayama, H.; Yun, Y.; Shirakawa, M. Multi-objective Model Predictive Control. Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems; Ehrgott, M.; Naujoks, B.; Stewart, T.J.; Wallenius, J., Eds. Springer Berlin Heidelberg, 2010, pp. 277–287.

143. Maester, J.M.; Muñoz de la Peña, D.; Camacho, E.F. Distributed model predictive control based on a cooperative game. *Optimal Control Applications and Methods* **2011**, *32*, 153–176.

144. Gambier, A. MPC and PID control based on Multi-Objective Optimization. American Control Conference, 2008, pp. 4727–4732.

145. Fonseca, C.M.M. Multiobjective genetic algorithms with application to control engineering problems. PhD thesis, University of Sheffield, 1995.

146. Herreros, A.; Baeyens, E.; Perán, J.R. MRCD: a genetic algorithm for multiobjective robust control design. *Engineering Applications of Artificial Intelligence* **2002**, *15*, 285–301.

147. Ben Aicha, F.; Bouani, F.; Ksouri, M. Automatic Tuning of GPC synthesis parameters based on Multi-Objective Optimization. XIth International Workshop on Symbolic and Numerical Methods, Modeling and Applications to Circuit Design (SM2ACD), 2010, pp. 1–5.

148. Krüger, M.; Witting, K.; Trächtler, A.; Dellnitz, M. Parametric Model-Order Reduction in Hierarchical Multiobjective Optimization of Mechatronic Systems. Proceedings of the 18th IFAC World Congress 2011, Milano, Italy. Elsevier Oxford, 2011, Vol. 18, pp. 12611–12619.

149. Hernández, C.; Naranjani, Y.; Sardahi, Y.; Liang, W.; Schütze, O.; Sun, J.Q. Simple cell mapping method for multi-objective optimal feedback control design. *International Journal of Dynamics and Control* **2013**, *1*, 231–238.

150. Xiong, F.R.; Qin, Z.C.; Xue, Y.; Schütze, O.; Ding, Q.; Sun, J.Q. Multi-objective optimal design of feedback controls for dynamical systems with hybrid simple cell mapping algorithm. *Communications in Nonlinear Science and Numerical Simulation* **2014**, *19*, 1465–1473.

151. Kobilarov, M. Discrete Geometric Motion Control of Autonomous Vehicles. PhD thesis, University of Southern California, 2008.

152. Flaßkamp, K.; Ober-Blöbaum, S.; Kobilarov, M. Solving optimal control problems by using inherent dynamical properties. *Proceedings of Applied Mathematics and Mechanics* **2010**, *10*, 577–578.

153. Eckstein, J.; Peitz, S.; Schäfer, K.; Friedel, P.; Köhler, U.; Hessel-von Molo, M.; Ober-Blöbaum, S.; Dellnitz, M. A Comparison of two Predictive Approaches to Control the Longitudinal Dynamics of Electric Vehicles. *Procedia Technology* **2016**, *26*, 465–472.

154. Wojsznis, W.; Mehta, A.; Wojsznis, P.; Thiele, D.; Blevins, T. Multi-objective optimization for model predictive control. *ISA Transactions* **2007**, *46*, 351–361.

155. Kerrigan, E.C.; Maciejowski, J.M. Designing model predictive controllers with prioritised constraints and objectives. IEEE International Symposium on Computer Aided Control System Design (CACSD), 2002, pp. 33–38.

156. Scherer, C.; Gahinet, P.; Chilali, M. Multiobjective Output-Feedback Control via LMI Optimization. *IEEE Transactions on Automatic Control* **1997**, *42*, 896–911.

157. Zambrano, D.; Camacho, E.F. Application of MPC with multiple objective for a solar refrigeration plant. IEEE International Conference on Control applications, 2002, pp. 1230–1235.

158. Porfírio, C.R.; Almeida Neto, E.; Odloak, D. Multi-model predictive control of an industrial C3/C4 splitter. *Control Engineering Practice* **2003**, *11*, 765–779.

159. Pedersen, G.K.M.; Yang, Z. Multi-objective PID-controller tuning for a magnetic levitation system using NSGA-II. Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation (GECCO), 2006, pp. 1737–1744.

160. Li, S.; Li, K.; Rajamani, R.; Wang, J. Model Predictive Multi-Objective Vehicular Adaptive Cruise Control. *IEEE Transactions on Control Systems Technology* **2011**, *19*, 556–566.

161. Hu, J.; Zhu, J.; Lei, G.; Platt, G.; Dorrell, D.G. Multi-objective model-predictive control for high-power converters. *IEEE Transactions on Energy Conversion* **2013**, *28*, 652–663.

162. Núñez, A.; Cortés, C.E.; Sáez, D.; De Schutter, B.; Gendreau, M. Multiobjective model predictive control for dynamic pickup and delivery problems. *Control Engineering Practice* **2014**, *32*, 73–86.

163.   Peitz, S.; Gräler, M.; Henke, C.; Hessel-von Molo, M.; Dellnitz, M.; Trächtler, A. Multiobjective Model Predictive Control of an Industrial Laundry. *Procedia Technology* **2016**, *26*, 483–490.

164.   Schütze, O.; Lara, A.; Coello Coello, C.A. On the influence of the Number of Objectives on the Hardness of a Multiobjective Optimization Problem. *IEEE Transactions on Evolutionary Computation* **2011**, *15*, 444–455.

165.   Fleming, P.J.; Purshouse, R.C.; Lygoe, R.J. Many-Objective Optimization: An Engineering Design Perspective. International Conference on Evolutionary Multi-Criterion Optimization, 2005, pp. 14–32.

166.   Kukkonen, S.; Lampinen, J. Ranking-Dominance and Many-Objective Optimization. IEEE Congress on Evolutionary Computation, 2007, pp. 3983–3990.

167.   Bader, J.; Zitzler, E. HypE : An Algorithm for Fast Hypervolume-Based Many-Objective Optimization. *Evolutionary Computation* **2011**, *19*, 45–76.

168.   Purshouse, R.C.; Fleming, P.J. On the Evolutionary Optimization of Many Conflicting Objectives. *IEEE Transactions on Evolutionary Computation* **2007**, *11*, 770–784.

169.   Ishibuchi, H.; Tsukamoto, N.; Nojima, Y. Evolutionary Many-Objective Optimization: A short Review. *Proceedings of the 2008 IEEE Congress on Evolutionary Computation* **2008**, pp. 2419–2426.

170.   Von Lücken, C.; Barán, B.; Brizuela, C. A survey on multi-objective evolutionary algorithms for many-objective problems. *Computational Optimization and Applications* **2014**, *58*, 707–756.

171.   Yang, S.; Li, M.; Liu, X.; Zheng, J. A Grid-Based Evolutionary Algorithm for Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation* **2013**, *17*, 721–736.

172.   Li, B.; Li, J.; Tang, K.; Yao, X. Many-Objective Evolutionary Algorithms: A Survey. *ACM Computing Surveys (CSUR)* **2015**, *48*, 13.

173.   Alves, M.J.; Clímaco, J. A review of interactive methods for multiobjective integer and mixed-integer programming. *European Journal of Operational Research* **2007**, *180*, 99–115.

174.   Monz, M.; Küfer, K.H.; Bortfeld, T.R.; Thieke, C. Pareto navigation - Algorithmic foundation of interactive multi-criteria IMRT planning. *Physics in Medicine and Biology* **2008**, *53*, 985–998.

175.   Eskelinen, P.; Miettinen, K.; Klamroth, K.; Hakanen, J. Pareto navigator for interactive nonlinear multiobjective optimization. *OR Spectrum* **2010**, *32*, 211–227.

176.   Cuate, O.; Lara, A.; Schütze, O. A Local Exploration Tool for Linear Many Objective Optimization Problems. 13th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), 2016.

177.   Cuate, O.; Derbel, B.; Liefooghe, A.; Talbi, E.G. An Approach for the Local Exploration of Discrete Many Objective Optimization Problems. In *Evolutionary Multi-Criterion Optimization: 9th International Conference, EMO 2017*; Trautmann, H.; Rudolph, G.; Klamroth, K.; Schütze, O.; Wiecek, M.; Jin, Y.; Grimme, C., Eds.; Springer International Publishing, 2017; pp. 135–150.

178.   Martin, A.; Schütze, O. Pareto Tracer: a predictor-corrector method for multi-objective optimization problems. *Engineering Optimization* **2018**, *50*, 516–536.

179.   Brockhoff, D.; Zitzler, E. Objective Reduction in Evolutionary Multiobjective Optimization: Theory and Applications. *Evolutionary Computation* **2009**, *17*, 135–166.

180.   Gu, F.; Liu, H.L.; Cheung, Y.M. A Fast Objective Reduction Algorithm Based on Dominance Structure for Many Objective Optimization. Simulated Evolution and Learning; Shi, Y.; Tan, K.C.; Zhang, M.; Tang, K.; Li, X.; Zhang, Q.; Tan, Y.; Middendorf, M.; Jin, Y., Eds. Springer International Publishing, 2017, pp. 260–271.

181.   Deb, K.; Saxena, D.K. On Finding Pareto-Optimal Solutions Through Dimensionality Reduction for Certain Large-Dimensional Multi-Objective Optimization Problems. *Kangal report* **2005**, *2005011*.

182.   Saxena, D.K.; Duro, J.A.; Tiwari, A.; Deb, K.; Zhang, Q. Objective reduction in many-objective optimization: Linear and nonlinear algorithms. *IEEE Transactions on Evolutionary Computation* **2013**, *17*, 77–99.

183.   Li, Y.; Liu, H.; Gu, F. An objective reduction algorithm based on hyperplane approximation for many-objective optimization problems. *2016 IEEE Congress on Evolutionary Computation, CEC 2016* **2016**, pp. 2470–2476.

184.   Wang, H.; Yao, X. Objective reduction based on nonlinear correlation information entropy. *Soft Computing* **2016**, *20*, 2393–2407.

185.   Bandyopadhyay, S.; Mukherjee, A. An Algorithm for Many-Objective Optimization With Reduced Objective Computations: A Study in Differential Evolution. *IEEE Transactions on Evolutionary Computation* **2015**, *19*, 400–413.

186.   Wang, H.; Jiao, L.; Shang, R.; He, S.; Liu, F. A Memetic Optimization Strategy Based on Dimension Reduction in Decision Space. _Evolutionary Computation_ **2010**, _23_, 69–100.

187.   Singh, H.K.; Isaacs, A.; Ray, T. A Pareto Corner Search Evolutionary Algorithm and Dimensionality Reduction in Many-Objective Optimization Problems. _IEEE Transactions on Evolutionary Computation_ **2011**, _15_, 539–556.

188.   Mueller-Gritschneder, D.; Graeb, H.; Schlichtmann, U. A successive approach to compute the bounded pareto front of practical multiobjective optimization problems. _SIAM Journal on Optimization_ **2009**, _20_, 915–934.

189.   de S. Motta, R.; Afonso, S.M.B.; Lyra, P.R.M. A modified NBI and NC method for the solution of N-multiobjective optimization problems. _Structural and Multidisciplinary Optimization_ **2012**, _46_, 239–259.

190.   Gebken, B.; Peitz, S.; Dellnitz, M. On the hierarchical structure of Pareto critical sets. _arXiv:1803.06864_ **2018**.

191.   Budišić, M.; Mohr, R.; Mezić, I. Applied Koopmanism. _Chaos_ **2012**, _22_.

192.   Rowley, C.W.; Mezić, I.; Bagheri, S.; Schlatter, P.; Henningson, D.S. Spectral analysis of nonlinear flows. _Journal of Fluid Mechanics_ **2009**, _641_, 115–127.

193.   Schmid, P.J. Dynamic mode decomposition of numerical and experimental data. _Journal of Fluid Mechanics_ **2010**, _656_, 5–28.

194.   Tu, J.H.; Rowley, C.W.; Luchtenburg, D.M.; Brunton, S.L.; Kutz, J.N. On Dynamic Mode Decomposition: Theory and Applications. _Journal of Computational Dynamics_ **2014**, _1_, 391–421.

195.   Proctor, J.L.; Brunton, S.L.; Kutz, J.N. Dynamic mode decomposition with control. _SIAM Journal on Applied Dynamical Systems_ **2015**, _15_, 142–161.

196.   Korda, M.; Mezić, I. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. _arXiv:1611.03537_ **2016**.

197.   Peitz, S.; Klus, S. Koopman operator-based model reduction for switched-system control of PDEs. _arXiv:1710:06759_ **2017**.

198.   Peitz, S. Controlling nonlinear PDEs using low-dimensional bilinear approximations obtained from data. _arXiv:1801.06419_ **2018**.

199.   Abraham, I.; De La Torre, G.; Murphey, T.D. Model-Based Control Using Koopman Operators. _arXiv:1709.01568_ **2017**.

200.   Hanke, S.; Peitz, S.; Wallscheid, O.; Klus, S.; Böcker, J.; Dellnitz, M. Koopman Operator Based Finite-Set Model Predictive Control for Electrical Drives. _arXiv:1804.00854_ **2018**.

201.   Duriez, T.; Brunton, S.L.; Noack, B.R. _Machine Learning Control – Taming Nonlinear Dynamics and Turbulence_; Springer, 2017.