POD-Based Multiobjective Optimal Control of Time-Variant Heat Phenomena

Stefan Banholzer, Eugen Makarov, and Stefan Volkwein

Abstract In the present paper, a multiobjective optimal control problem governed by a heat equation with time-dependent convection term and bilateral control constraints is considered. For computing Pareto optimal points and approximating the Pareto front, the reference point method is applied. As this method transforms the multiobjective optimal control problem into a series of scalar optimization problems, the method of proper orthogonal decomposition (POD) is introduced as an approach for model-order reduction. New strategies for efficiently updating the POD basis in the optimization process are proposed and tested numerically.

1 Introduction

Many optimization problems in applications can be formulated using several objective functions, which are conflicting with each other. This leads to the notion of multiobjective or multicriterial optimization problems; cf. [4, 9, 12]. One prominent example is given by an energy efficient heating, ventilation and air-conditioning (HVAC) operation of a building with conflicting objectives such as minimal energy consumption and maximal comfort; cf. [6, 8].

In this paper we apply the reference point method [11] in order to transform a bicriterial optimal control problem into a sequence of scalar-valued optimal control problems and solve them using well-known optimal control techniques; see [13]. We build on and extend previous results obtained in [2], where a linear convection-diffusion equation was considered. In addition, we allow the convection term to be time-dependent here.

By using the a-posteriori error estimate [2, Theorem 9] we develop a new strategy for updating the POD basis while computing the Pareto front such that the error stays always below a certain predefined threshold. In our numerical examples we

S. Banholzer · E. Makarov · S. Volkwein (✉)
University of Konstanz, Department of Mathematics and Statistics, Konstanz, Germany
e-mail: Stefan.Banholzer@uni-konstanz.de; Stefan.Volkwein@uni-konstanz.de

F. A. Radu et al. (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2017*, Lecture Notes in Computational Science and Engineering 126,
https://doi.org/10.1007/978-3-319-96415-7_83

compare the strategy with the simple basis extension algorithm in [2, Algorithm 3]. Moreover, we propose a method to choose an efficient initial number of POD basis functions.

The paper is organized in the following manner: In Sect. 2 we present the state equation and the bicriterial optimal control problem. The reference point method and how to apply it to the problem at hand is explained in Sect. 3. Moreover, the POD method is briefly introduced to gain a speed-up in the solution process. Section 4 contains the numerical experiments and in Sect. 5 we draw a conclusion.

2 Problem Formulation

The State Equation Let $\Omega \subset \mathbb{R}^d$ with $d \in \{2, 3\}$ be a bounded domain with Lipschitz-continuous boundary Γ. We choose m non-empty, pairwise disjoint subsets $\Omega_1, \dots, \Omega_m$ of the domain Ω. For a given end time $T > 0$, we set $Q := (0, T) \times \Omega$ and $\Sigma := (0, T) \times \Gamma$. The state equation is then given by the following diffusion-convection equation with homogeneous Neumann boundary conditions:

$$y_t(t,x) - \kappa \Delta y(t,x) + \beta(t,x) \cdot \nabla y(t,x) = \sum_{i=1}^{m} u_i(t)\chi_i(x) \qquad \text{in } Q, \tag{1a}$$

$$\frac{\partial y}{\partial \boldsymbol{n}}(t,x) = 0 \qquad \text{on } \Sigma, \tag{1b}$$

$$y(0,x) = y_0(x) \qquad \text{in } \Omega. \tag{1c}$$

In (1a) the constant $\kappa > 0$ is the diffusion coefficient and the time-dependent advection β is supposed to be in $L^\infty(Q; \mathbb{R}^d)$. Furthermore, the function χ_i is given by the characteristic function of the set Ω_i for all $i = 1, \dots, m$. For the control variable $u = (u_1, \dots, u_m)$ we assume $u \in U = L^2(0, T; \mathbb{R}^m)$. Finally in (1c), $y_0 \in H = L^2(\Omega)$ is a given initial temperature. To set the framework for the weak formulation of (1), we define the Hilbert space $V = H^1(\Omega)$ equipped with the standard inner product. The space

$$Y = W(0,T) = \big\{\phi \in L^2(0,T;V) \mid \phi_t \in L^2(0,T;V')\big\}$$

endowed with the canonical inner product is a Hilbert space; see, e.g. [3]. With similar arguments as in [1, Section 5.1] it is possible to show that for each tuple $(u, y_0) \in U \times H$ there is a unique weak solution $y \in Y$ of (1). Furthermore, the solution can be written as $y = \hat{y} + \mathcal{S}u$, where $\hat{y} \in Y$ is the weak solution of (1) for the pair $(0, y_0)$ and the linear operator $\mathcal{S} : U \to Y$ is given such that $\mathcal{S}u$ is the weak solution of (1) to the pair $(u, 0)$.

The Bicriterial Optimal Control Problem For a given desired temperature $y_Q \in L^2(0,T;H)$ we introduce the cost functional

$$J : Y \times U \to \mathbb{R}^2, \quad J(y,u) = \Big(\tfrac{1}{2} \left\| y - y_Q \right\|^2_{L^2(0,T;H)}, \tfrac{1}{2} \left\| u \right\|^2_{L^2(0,T;\mathbb{R}^m)} \Big)^\top.$$

Defining the set $U_{\mathsf{ad}} = \{u \in U \mid u_a \leq u \leq u_b \text{ in } [0,T]\}$ for given $u_a, u_b \in U$ with $u_a \leq u_b$ in $[0,T]$, the bicriterial optimal control problem reads

$$\min J(y,u) \quad \text{s.t.} \quad (y,u) \in \big\{(\tilde{y},\tilde{u}) \in Y \times U_{\mathsf{ad}} \mid \tilde{y} = \hat{y} + \mathcal{S}\tilde{u}\big\}. \tag{2}$$

Since $\mathcal{S}$ is well-defined, we define the reduced cost function $\hat{J} : U \to \mathbb{R}^2$, $\hat{J}(u) = J(\hat{y} + \mathcal{S}u, u)$ and investigate the reduced formulation of (2) in this paper:

$$\min \hat{J}(u) \quad \text{s.t.} \quad u \in U_{\mathsf{ad}}. \tag{3}$$

Problem (3) involves the minimization of a vector-valued function with two objectives. This is done by using the concept of *Pareto optimality*; cf. [4].

Definition 1 The point $\bar{u} \in U_{\mathsf{ad}}$ is called *Pareto optimal* for (3) if there is no other control $u \in U_{\mathsf{ad}} \setminus \{\bar{u}\}$ with $\hat{J}_i(u) \leq \hat{J}_i(\bar{u})$, $i = 1, 2$, and $\hat{J}(u) \neq \hat{J}(\bar{u})$.

3 The Reference Point Method

The theoretical and numerical aim in solving a bicriterial optimization problem is to get an approximation of the *Pareto set* and the *Pareto front*, respectively, which are given by

$$P_s = \big\{u \in U_{\mathsf{ad}} \mid u \text{ is Pareto optimal}\big\} \subset U \quad \text{and} \quad P_f = \hat{J}(P_s) \subset \mathbb{R}^2.$$

The scalarization method, in which the bicriterial function is transformed into a scalar function and then minimized using well-known techniques from scalar optimization, is one of the most popular approaches to tackle this problem, see e.g. [5, 9, 12]. The idea is that by choosing different scalarizations, both the Pareto set and the Pareto front can be approximated. One particular scalarization method is the (Euclidean) *reference point method*, which was previously used in [10, 11]. Given a reference point $z \in P_f + \mathbb{R}^2_{\leq} = \{z + x \mid z \in \mathcal{P}_f \text{ and } x \in \mathbb{R}^2_{\leq}\}$ the distance function

$$F_z : U \to \mathbb{R}, \quad F_z(u) = \tfrac{1}{2} \big(\hat{J}_1(u) - z_1\big)^2 + \tfrac{1}{2} \big(\hat{J}_2(u) - z_2\big)^2$$

measures the Euclidean distance between $\hat{J}(u)$ and z for a given $u \in U$. The idea is that by solving the minimization problem

$$\min F_z(u) \quad \text{s.t.} \quad u \in U_{\mathsf{ad}}, \tag{4}$$

we get a Pareto optimal point for (3). The following theorem, which is taken from [1, Theorem 3.35], guarantees this property for the problem at hand.

Theorem 2 *Let $z \in P_f + \mathbb{R}^2_{\leq}$ be a reference point. Then* (4) *has a unique solution $\bar{u} \in U_{\mathsf{ad}}$, which is Pareto optimal for* (3).

The algorithmic approach to approximate P_f, which we consider in this paper, first computes the two boundary points of the Pareto front. These are given by the minimizers of $\hat{J}_1$ and $\hat{J}_2$, respectively. In our case we have to regularize the minimization of $\hat{J}_1$ because $\hat{J}_1$ is only strictly, but not strongly convex. Therefore, we minimize $\hat{J}_1 + \alpha \hat{J}_2$ with a small weight $0 < \alpha \ll 1$. We always choose the minimizer of $\hat{J}_1$ as a starting point. Given a Pareto optimal point the algorithm generates a new reference point following P_f from top to bottom, and then solves the respective reference point problem. This procedure is repeated until the end of P_f is reached. The exact scheme for computing the reference points along with a more detailed description of the algorithm can be found in [1, Section 3.4] and [2, Section 6].

When implementing this algorithm, (4) has to be solved repeatedly for different reference points $z \in P_f + \mathbb{R}^2_{\leq}$. Each solve of (4) requires multiple solves of (1) and an adjoint equation (cf. [13, Section 3.6]) which is often computationally too costly when using a standard Finite Element (FE) method. Thus, it is reasonable to apply model-order reduction to reduce the computational effort. We use the well-known POD method; cf. [7]. In [1, 2] the procedure for our problem at hand is explained. Here, we just want to introduce some notations: Given a POD basis $\{\psi_i\}_{i=1}^{\ell}$ of rank ℓ, we define the set $V^\ell = \operatorname{span}\{\psi_1, \ldots, \psi_\ell\}$ and the solution operator $\mathcal{S}^\ell : U \to H^1(0, T; V^\ell)$ of the POD solution of the u-dependent part of the state equation. The POD approximation of $\hat{J}$ is defined as $\hat{J}^\ell(u) = J(\hat{y} + \mathcal{S}^\ell u, u)$. Then, (4) is replaced by

$$\min F_z^\ell(u) = \tfrac{1}{2}\left(\hat{J}_1^\ell(u) - z_1\right)^2 + \tfrac{1}{2}\left(\hat{J}_2^\ell(u) - z_2\right)^2 \quad \text{s.t.} \quad u \in U_{\mathsf{ad}} \tag{5}$$

with $\hat{J}_2^\ell(u) = \hat{J}_2(u)$. For a given reference point $z \in \mathcal{P}_f + \mathbb{R}^2_{\leq}$ we denote the optimal solutions to (4) and (5) by $\bar{u}_z$ and $\bar{u}_z^\ell$, respectively.

4 Numerical Results

In our numerical tests we consider the bicriterial optimal control problem presented in Sect. 2. We have $\Omega = (0, 1)^2 \subset \mathbb{R}^2$ and choose $T = 1$. The diffusion parameter is given by $\kappa = 0.5$ and for the convection term in (1a) we use $\beta(t, x) = c_b \tilde{\beta}(t, x)$ for all $(t, x) \in Q$, where $\tilde{\beta}$ is a non-stationary solution of a Navier-Stokes equation and $c_b \geq 0$ is a parameter to control the strength of the convection; cf. Fig. 1. We impose a floor heating of the whole room with $m = 4$ uniformly distributed heaters in the domains $\Omega_1 = (0, 0.5)^2$, $\Omega_2 = (0, 0.5) \times (0.5, 1)$, $\Omega_3 = (0.5, 1) \times (0, 0.5)$ and

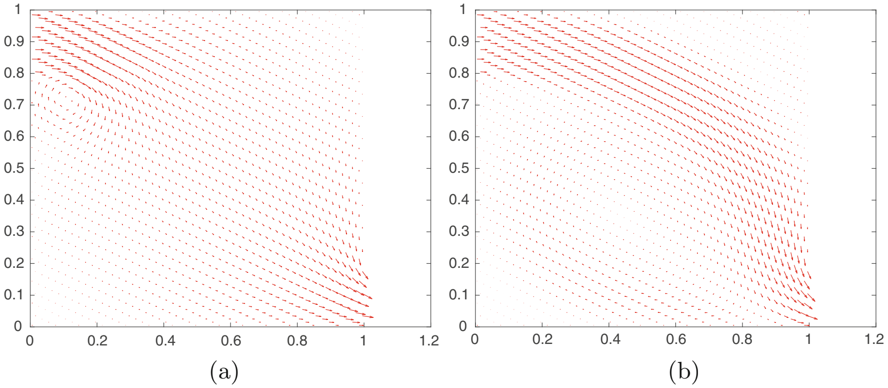

Fig. 1 Time-dependent convection $\beta(t, x)$ at two time instances. (**a**) $t = 0.01$. (**b**) $t = 0.5$

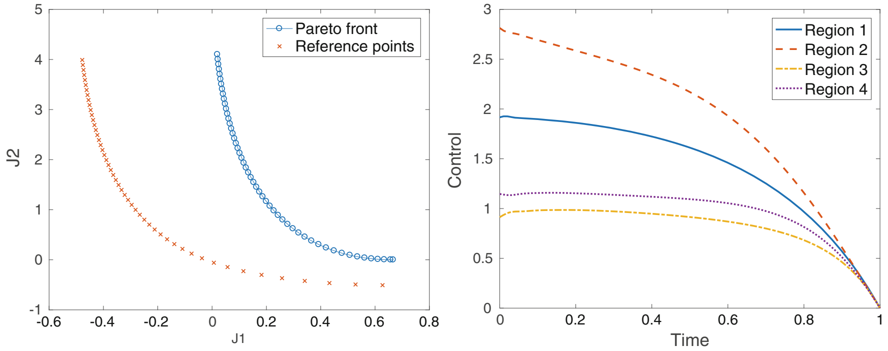

Fig. 2 Test 1: Pareto front (left); optimal control $\bar{u}^1$ corresponding to P^1 (right)

$\Omega_4 = (0.5, 1)^2$. The bilateral control constraints are $u_a = 0$ and $u_b = 3$. Finally, we choose $y_0 = 16$ and $y_Q(t, x) = 16 + 2t$ for all $(t, x) \in Q$. All computations were carried out on a MacBook Pro 13 (middle 2012) with 2.5 GHz Intel Core i5 and 4 GB RAM.

Test 1 We solve (3) for $c_b = 1$. Then, P_f is smoothly approximated by 52 Pareto optimal points; cf. Fig. 2. Hereby, P_f ranges from $P^1 = (0.0199, 4.1)$, which is computed with the weighted-sum method with weight $\alpha = 0.02$, to $P^{52} = (0.6667, 0)$. Thus, the desired temperature can be achieved quite closely in the upper part of P_f. The four optimal controls for P^1 can be seen in right plot of Fig. 2. As in the case of a time-independent convection term all four controls adapt to the air flow, which goes from the top left corner of the room to the right bottom corner by using different heating strategies. Furthermore, one can observe a slightly wavy behaviour of the optimal controls at the beginning. This is due to the temporal changes in the dynamics of the system caused by a vortex moving over time from the top left corner to the middle of the room. Another interesting aspect is that in the comparison to the time-independent case (by taking average over the time) the system with time-dependent convection only needs about 10% more computation time although the latter case adds dynamics to the optimal control problem, which are more difficult to handle numerically.

Test 2 Now we use an adaptive POD basis extension algorithm, which was introduced in [2, Algorithm 3], in order to investigate the influence of the time-dependent convection term on the number of needed POD basis functions. As a measure for the error between u_z^ℓ and $\bar{u}_z^\ell$ induced by the POD approximation we use the a-posteriori error estimate [2, Theorem 9] and set the upper bound of the acceptable error to $\mu = 4 \cdot 10^{-4}$, as well as the initial number of POD basis functions to $\ell_{init} = 5$. Choosing the minimizer of $\hat{J}_1$ as starting point we observe that 24 POD basis functions are needed to compute the whole Pareto front in the desired approximation quality. Thereby, all 19 basis extensions are conducted on P^2. For comparison, in [2] was shown that for a very similar problem with a time-independent convection term only 15 POD basis functions are sufficient to achieve the same quality. This is due to the fact that a time-dependency of a convection adds more complex dynamics to the system which are hard to capture by using only one fixed POD basis. To tackle this problem we propose a POD basis improving strategy. Algorithm 1 shows the routine for computing the n-th Pareto optimal point. In this paper we investigate two strategies for determining ℓ after each basis update. In the first case, we set $\ell = \ell_{min}$. In the second case, we choose ℓ by observing the convergence rates in the control space; cf. [7, Theorem 1.49]. Namely, by choosing $\ell \in [\ell_{min}, \ell_{max}]$ such that

$$\sum_{i=\ell+1}^{\ell_{max}} \lambda_i \|\psi_i - \mathcal{P}_H^\ell \psi_i\|_{H^1(\Omega)}^2 < \varepsilon \tag{6}$$

for an $\varepsilon < \mu$ holds, where $\mathcal{P}_H^\ell$ denotes the H-orthogonal projection onto V^ℓ, $\{\psi_i\}_{i\in\mathbb{N}}$ is a POD basis and $\{\lambda_i\}_{i\in\mathbb{N}}$ are the corresponding eigenvalues. The results

Algorithm 1 POD basis update algorithm

```
Require: threshold $\mu > 0$, $0 < \ell_{min} < \ell_{max}$;
Set check = 0;
while check = 0 do
    Solve (5) with reference point $z^{(n)}$;
    Compute the a-posteriori error estimate $\mu_{apost}$ for the controls;
    if $\mu_{apost} < \mu$ then
        Set check = 1;
    else
        if $\ell < \ell_{max} - 1$ then
            Set $\ell = \ell + 2$;
        else
            Solve (4) for $\bar{u}_z$ with reference point $z^{(n)}$ and starting point $\bar{u}_z^\ell$ ;
            Compute new POD basis by using $\bar{u}_z$;
            Choose $\ell \in [\ell_{min}, \ell_{max}]$ and set check = 1.
```

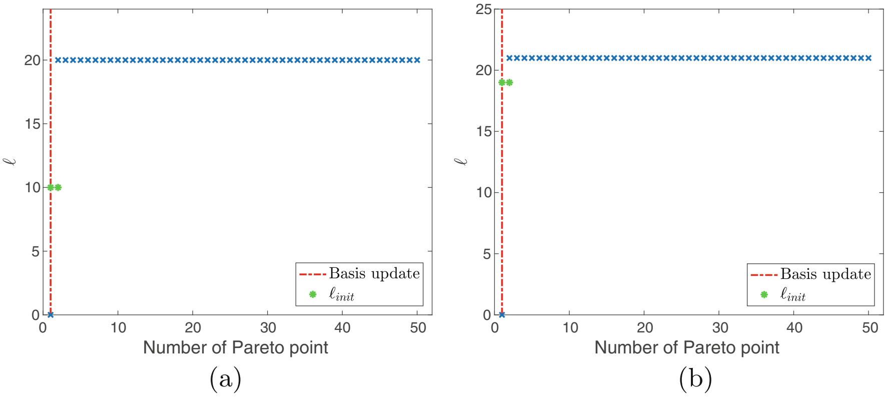

Fig. 3 Number of used POD basis functions in Algorithm 1 using $\varepsilon = 0.5 \cdot 10^{-2}\mu$. (**a**) $\ell = 10$ fixed. (**b**) $6 \le \ell \le 22$ adaptive

Table 1 Test 2: Results for $\varepsilon = 0.5 \cdot 10^{-2}\mu$

	CPU time	#Basis extensions	#Basis updates
Full system	240.25	–	–
Basis extension with $\ell_{init} = 5$	51.07	19	–
Algorithm 1 with fixed $\ell = 10$	49.53	11	1
Algorithm 1 with adaptive $\ell \in [6, 22]$	44.82	2	1
Algorithm 1 with adaptive $\ell \in [6, 20]$	59.51	0	6

for $\varepsilon = 5 \cdot 10^{-3}\mu$ and $\ell_{max} = 22$ are presented in Fig. 3 and Table 1. The value of ℓ_{min} is set to 10 and to 6 by the first and second strategy, respectively. Using the second strategy Algorithm 1 yields the best results with respect to the CPU time. Thus, (6) estimates quite well how many POD basis functions would have been necessary in order to compute the current Pareto point. As a result only 2 basis extensions on the points P^2 and P^3 are needed. Using the first strategy, 11 basis extensions are necessary as 10 POD basis function are not enough even after a basis update. Hence, a lot of avoidable basis extensions are done. In both cases one basis update is conducted. However, the performance of the Algorithm 1 depends strongly on the choice of ℓ_{max}. Decreasing ℓ_{max} to 20 increases the CPU time by 33% as 6 basis updates are needed in this case. Furthermore, ε has to be chosen appropriately to avoid unnecessary basis extensions in the second case.

Test 3 Now we increase the strength of the convection term c_b to 2 and run the basis extension algorithm with the same settings. Surprisingly, in the basis extension algorithm [2, Algorithm 3] only 25 POD basis functions are needed to compute the whole Pareto front, although there are significant changes in the behaviour of the controls. However, as expected, increasing the strength of the convection term increases heavily the number of basis updates in Algorithm 1 for the same values of ℓ_{max} and thus the computation time.

5 Conclusion

In the present paper we show how including a time-dependent advection term into the state equation influences the results of the bicriterial optimal control problem (2). As expected the time-dependence adds dynamics to the system which cannot be captured that easily by a single POD basis. Therefore, by introducing a new POD update strategy we are able to save about 15% of the CPU time in comparison to the basis extension algorithm in [2, Algorithm 3].

Acknowledgements S. Banholzer gratefully acknowledges support by the German DFG-Priority Program 1962 and by the Landesgraduiertenförderung of Baden-Württemberg.

References

1. S. Banholzer, POD-Based Bicriterial Optimal Control of Convection-Diffusion Equations, Master thesis, University of Konstanz, Department of Mathematics and Statistics, 2017, see http://nbn-resolving.de/ `urn:nbn:de:bsz:352-0-421545`
2. S. Banholzer, D. Beermann, S. Volkwein, POD-based error control for reduced-order bicriterial PDE-constrained optimization. Annu. Rev. Control **44**, 226–237 (2017)
3. R. Dautray, J.-L. Lions, *Mathematical Analysis and Numerical Methods for Science and Technology. Volume 5: Evolution Problems I* (Springer, Berlin, 2000)
4. M. Ehrgott, *Multicriteria Optimization* (Springer, Berlin, 2005)
5. G. Eichfelder, *Adaptive Scalarization Methods in Multiobjective Optimization* (Springer, Berlin, 2008)
6. K.F. Fong, V.I. Hanby, T.-T. Chow, HVAC system optimization for energy management by evolutionary programming. Energy Build. **38**, 220–231 (2006)
7. M. Gubisch, S. Volkwein, Proper orthogonal decomposition for linear-quadratic optimal control, in *Model Reduction and Approximation: Theory and Algorithms*, ed. by P. Benner, A. Cohen, M. Ohlberger, K. Willcox (SIAM, Philadelphia, 2017), pp. 5–66
8. A. Kusiak, F. Tang, G. Xu, Multi-objective optimization of HVAC system with an evolutionary computation algorithm. Energy **36**, 2440–2449 (2011)
9. K. Miettinen, *Nonlinear Multiobjective Optimization*. International Series in Operations Research and Management Science (Springer, Berlin, 1998)
10. S. Peitz, S. Oder-Blöbaum, M. Dellnitz, Multiobjective optimal control methods for fluid flow using reduced order modeling, in *24th Congress of Theoretical and Applied Mechanics (ICTAM)*, Montreal, Canada, 21–26 August 2016, see http://arxiv.org/pdf/1510.05819v2.pdf
11. C. Romaus, J. Böcker, K. Witting, A. Seifried, O. Znamenshchykov, Optimal energy management for a hybrid energy storage system combining batteries and double layer capacitors, in *IEEE Energy Conversion Congress and Exposition*, San Jose, 2009, pp. 1640–1647
12. W. Stadler, *Multicriteria Optimization in Engineering and in the Sciences* (Plenum Press, New York, 1988)
13. F. Tröltzsch, *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*, 2nd edn. (American Mathematical Society, Providence, 2010)