This document contains the **post-print pdf-version** of the refereed paper:

“Fast Pareto set generation for nonlinear optimal control problems with multiple objectives”

by *Filip Logist, Boris Houska, Moritz Diehl, and Jan Van Impe*

which has been archived on the university repository Lirias (<https://lirias.kuleuven.be/>) of the Katholieke Universiteit Leuven.

The content is identical to the content of the published paper, but without the final typesetting by the publisher.

When referring to this work, please cite the full bibliographic info:

F. Logist, B. Houska, M. Diehl, Jan F. Van Impe (2010). Fast Pareto set generation for nonlinear optimal control problems with multiple objectives, Structural and Multidisciplinary Optimization, 42, 591-603.

The journal and the original published paper can be found at:

<http://link.springer.com/journal/158>

<http://link.springer.com/article/10.1007/s00158-010-0506-x>

The corresponding author can be contacted for additional info.

Conditions for open access are available at:

<http://www.sherpa.ac.uk/romeo/>

Structural and multidisciplinary optimization manuscript No. (will be inserted by the editor)
--

Fast Pareto set generation for nonlinear optimal control problems with multiple objectives

Filip Logist · Boris Houska · Moritz Diehl · Jan Van Impe

Received: 27 October 2009 / Revised: 12 March 2010 / Accepted: 12 March 2010

Abstract Many practical engineering problems involve the determination of optimal control trajectories for given multiple and conflicting objectives. These conflicting objectives typically give rise to a set of Pareto optimal solutions. To enhance real-time decision making efficient approaches are required for determining the Pareto set in a fast and accurate way. Hereto, the current paper integrates efficient multiple objective scalarisation strategies (e.g., Normal Boundary Intersection and Normalised Normal Constraint) with fast deterministic approaches for dynamic optimisation (e.g., Single and Multiple Shooting). All techniques have been implemented as an easy-to-use add-on module of the automatic control and dynamic optimisation toolkit ACADO (both freely available at www.acadotoolkit.org). Several algorithmic synergies (e.g., hot-start initialisation strategies) are exploited for an additional speed-up. The features of ACADO Multi-Objective are discussed and its use is illustrated on different multiple objective optimal control problems arising in several engineering disciplines.

Keywords multiobjective optimization · optimal control · dynamic optimization · Normalized Normal Constraint · Normal Boundary Intersection

F. Logist and J. Van Impe
BioTeC & Optimization in Engineering Center (OPTEC)
Department of Chemical Engineering, K.U.Leuven
W. de Croylaan 46, B-3001 Leuven, Belgium
Tel: +32.16.32.14.66; Fax: +32.16.32.29.91
E-mail: {filip.logist,jan.vanimpe}@cit.kuleuven.be

B. Houska and M. Diehl
SCD & Optimization in Engineering Center (OPTEC)
Department of Electrical Engineering, K.U.Leuven
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium
Tel: +32.16.32.18.84; Fax: +32.16.32.85.39
E-mail: {boris.houska,moritz.diehl}@esat.kuleuven.be

PACS Numerical optimization · Control systems · Computer software

1 Introduction

Optimal control or *dynamic optimisation* problems, i.e., finding an optimal time-varying control trajectory for a dynamic system described by differential equations, are encountered in many engineering disciplines. Typical examples concern the design of, e.g., (i) optimal rocket trajectories in aerospace engineering, (ii) optimal robot paths in mechanical engineering, (iii) optimal power management of fuel cells in electrical engineering, (iv) optimal reactor feeding rates in (bio)chemical engineering or (v) optimal drug administration in (bio)medical engineering.

To support and enhance decision-making in practice, systematic mathematical techniques and software tools are required to present different alternatives (i.e., Pareto optimal solutions) to the decision maker (e.g., the design or control engineer), from which he/she can pick one based on his/her preferences. In the literature two classes of methods for systematically generating the Pareto set exist (see, e.g., the review by Marler and Arora (2004)): (i) *scalarisation methods* (e.g., Miettinen (1999)) converting the multiple objective optimisation problem (MOOP) into a series of parametric single objective optimisation problems (SOOPs) (e.g., weighted sum, . . .), and (ii) *vectorisation methods* tackling directly the MOOP (e.g., stochastic evolutionary algorithms (Deb 2001)).

The *vectorisation* techniques are most often easily implemented, can flexibly be coupled to any process simulator, and they are generally regarded as global optimisation approaches. Hence, a large number of suc-

successful applications has been reported over the years (see also a review of chemical engineering applications by Bhaskar et al (2000)). Unfortunately, these methods also exhibit certain restrictions. The repeated evaluation of the objectives is required (and, thus, also the repeated simulation of the underlying models), which can become time consuming. Other constraints than simple bounds are hard to cope with in an accurate way. And finally, due to the stochastic nature of the search procedures, the dimension of the search space has to be kept rather low. These last two features make these techniques not always suitable for multiple objective optimal control problems (MOOCs), where nonlinear constraints are often present and an accurate determination of the control profile is desired.

Alternatively, *scalarisation* techniques in general do not suffer from these drawbacks since efficient deterministic optimisation approaches can be exploited to find a (local) optimum for the different possibly large-scale SOOPs. The most often exploited scalarisation technique is the weighted sum (WS) of the different individual objectives. However, it suffers from several intrinsic drawbacks. A uniform variation of the weights does not necessarily lead to an even spread on the Pareto front, and points in non-convex parts of the Pareto front cannot be obtained (Das and Dennis 1997). To overcome these drawbacks several other scalarisation approaches have been developed, e.g., Normal Boundary Intersection (NBI) (Das and Dennis 1998), Normalised Normal Constraint (NNC) (Messac et al 2003; Messac and Mattson 2004) and Adaptive Weighted Sum (Kim and de Weck 2005, 2006).

Recently, a successful application of NBI and NNC for the multiple objective optimal control of (bio)chemical processes has been reported by Logist et al (2009). Building on this work, the aim of the current paper is to report on the development of a generic and easy-to-use toolbox for multiple objective dynamic optimisation integrating efficient scalarisation techniques with accurate simultaneous optimisation methods. The efficient solution of the SOOPs is crucial, because their number increases exponentially with the number of the objectives. Therefore, several advanced (re-)initialisation strategies have been implemented which allow to hot-start SOOPs (i.e., start from the previous solution) and significantly decrease the computation time.

The structure of the paper is as follows. In Section 2 the mathematical formulation of a general MOOC is introduced. The rationale behind the toolbox ACA-DO Multi-Objective is described in Section 3 while in Section 4 implementation details are provided. Several examples from different engineering disciplines are discussed in Section 5.

2 Mathematical formulation

In general, a multiple objective optimal control problem (MOOC) can be formulated as follows.

$$\min_{\mathbf{x}(\xi), \mathbf{u}(\xi), \mathbf{p}, \xi_f} \{J_1, \dots, J_m\} \quad (1)$$

subject to:

$$\frac{d\mathbf{x}}{d\xi} = \mathbf{f}(\mathbf{x}(\xi), \mathbf{u}(\xi), \mathbf{p}, \xi) \quad \xi \in [0, \xi_f] \quad (2)$$

$$\mathbf{0} = \mathbf{b}_c(\mathbf{x}(0), \mathbf{x}(\xi_f), \mathbf{p}) \quad (3)$$

$$\mathbf{0} \geq \mathbf{c}_p(\mathbf{x}(\xi), \mathbf{u}(\xi), \mathbf{p}, \xi) \quad (4)$$

$$\mathbf{0} \geq \mathbf{c}_t(\mathbf{x}(\xi_f), \mathbf{u}(\xi_f), \mathbf{p}, \xi_f) \quad (5)$$

Here, $\mathbf{x}$ are the state variables, while $\mathbf{u}$ and $\mathbf{p}$ denote the time-varying and time-constant control variables, respectively. The vector $\mathbf{f}$ represents the dynamic system equations (on the interval $\xi \in [0, \xi_f]$) with initial and terminal boundary conditions given by the vector $\mathbf{b}_c$. The vectors $\mathbf{c}_p$ and $\mathbf{c}_t$ indicate respectively path and terminal inequality constraints on the states and controls. Each individual objective function can consist of both Mayer and Lagrange terms.

$$J_i = h_i(\mathbf{x}(\xi_f), \mathbf{p}, \xi_f) + \int_0^{\xi_f} g_i(\mathbf{x}(\xi), \mathbf{u}(\xi), \mathbf{p}, \xi) d\xi \quad (6)$$

In view of conciseness all design freedoms are grouped as $\mathbf{y} = (\mathbf{x}(\xi), \mathbf{u}(\xi), \mathbf{p}, \xi_f)$ and the set of feasible solutions $\mathcal{S}$ is defined as all vectors $\mathbf{y}$ that satisfy Equations (2) to (5). Similarly, the vector of all individual objective functions is defined as $\mathbf{J}(\mathbf{y}) = [J_1(\mathbf{y}), \dots, J_m(\mathbf{y})]^T$.

3 Theoretical background

In the current section the background is sketched of the deterministic multiple objective and optimal control techniques that are implemented to efficiently and accurately solve MOOCs.

3.1 Scalarisation approaches for multiple objective optimisation problems

Contrary to single objective optimisation, typically no single solution exists in multiple objective optimisation. The concept of *Pareto optimality* is adopted as a criterion to determine the set of optimal solutions.

Definition: A point $\mathbf{y}^* \in \mathcal{S}$, is Pareto optimal iff there does not exist another point $\mathbf{y} \in \mathcal{S}$, such that $J_i(\mathbf{y}) \leq J_i(\mathbf{y}^*)$ for all i and $J_j(\mathbf{y}) < J_j(\mathbf{y}^*)$ for at least one objective function j .

In other words, a point is Pareto optimal if there exists no other point that improves at least one objective function without worsening another. The rationale behind scalarisation techniques is to convert the original MOOP into a (series of) parametric SOOPs, whose solution each time should yield one point of the Pareto set. Hence by consistently varying the parameter(s) an approximation of the Pareto set is obtained.

3.1.1 Weighted Sum (WS)

The most often employed scalarisation technique in practice is the combination of the different objectives into a convex weighted sum¹:

$$\min_{\mathbf{y} \in \mathcal{S}} J_{\text{ws}} = \sum_{i=1}^m w_i J_i(\mathbf{y}) \text{ with } w_i \geq 0 \text{ and } \sum_{i=1}^m w_i = 1 \quad (7)$$

with the weight vector $\mathbf{w} = [w_1, w_2, \dots, w_i, \dots, w_m]^T$. However, despite its simplicity, the weighted sum approach has as intrinsic drawbacks that (i) a uniform distribution of the weight vector does not necessarily result in an even spread on the Pareto front and (ii) that points in non-convex parts of the Pareto set cannot be obtained. Nevertheless, the weights can easily be related to price information, whenever available. In this case the Pareto set will allow to analyse the sensitivity with respect to price changes.

3.1.2 Normal Boundary Intersection (NBI)

Das and Dennis (1998) proposed this geometrically intuitive approach to mitigate the drawbacks of the WS. (For a geometric interpretation of NBI in a (normalised) bi-objective case, see Figure 1.) NBI first builds a plane in the objective space $\mathcal{J}_f$ which contains all convex combinations of the individual minima, i.e., the *convex hull of individual minima* (CHIM), and then constructs (quasi-)normal lines to this plane. The rationale behind the method is that the intersection between the (quasi-)normal from any point $\mathbf{J}_p$ on the CHIM, and the boundary of the feasible objective space closest to the *utopia point* $\mathbf{J}^*$ (i.e., the point which contains the minima of the individual objectives $J_i(\mathbf{y}_i^*)$) is expected to be Pareto optimal. Hereto, the multiple objective optimisation problem is reformulated as to maximise the distance λ from a point $\mathbf{J}_p$ on the CHIM along the quasi-normal through this point, without violating the original constraints. Technically, this requirement of lying on the quasi-normal introduces additional equality

¹ Although it is usually supposed that the weights are normalised, i.e., $\sum_{i=1}^m w_i = 1$ (Miettinen 1999), this constraint is in general not necessary.

Fig. 1 Schematic representation of NBI (left) and NNC (right) for a bi-objective problem.

constraints, resulting in the following formulation:

$$\max_{\mathbf{y} \in \mathcal{S}, \lambda} \lambda \quad \text{s.t.} : \quad \Phi \mathbf{w} - \lambda \Phi \mathbf{e} = \mathbf{J}(\mathbf{y}) - \mathbf{J}^* \quad (8)$$

where Φ is the $m \times m$ pay-off matrix in which the i -th column is $\mathbf{J}(\mathbf{y}_i^*) - \mathbf{J}^*$. Here, $\mathbf{y}_i^*$ is the minimiser of the i -th objective J_i and $\mathbf{J}^*$ is the *utopia point*. $\mathbf{w}$ is again a vector of 'weights' $\mathbf{w} = [w_1, w_2, \dots, w_i, \dots, w_m]^T$ such that $\sum_{i=1}^m w_i = 1$ with $w_i \geq 0$, and $\mathbf{e}$ is a vector containing all ones. Now, $\Phi \mathbf{w}$ describes a point $\mathbf{J}_p$ in the CHIM and $-\Phi \mathbf{e}$ defines the (quasi-)normal to the CHIM pointing towards the utopia point. When the points on the CHIM are selected with an equal spread (via a uniform distribution of the 'weight' vectors $\mathbf{w}$), an equal spread on the Pareto frontier in the objective space is expected.

3.1.3 Normalised Normal Constraint (NNC)

Messac et al (2003) employed ideas similar to NBI, but combined them with features of the ε -constraint method (Haimes et al 1971). (For a geometric interpretation of NNC in a (normalised) bi-objective case, see Figure 1.) This ε -constraint method minimises the single most important objective function J_k , while the $m - 1$ other objective functions are added as inequality constraints $J_i \leq \varepsilon_i$. These inequalities can be interpreted as hyperplanes reducing the feasible objective space. NNC starts with a normalisation of the objectives. Then it also constructs a plane through all (normalised) individual minima (called here, the *utopia hyperplane*). Afterwards, NNC minimises a selected (normalised) objective $\bar{J}_k$, given the original constraints, and while additionally reducing the feasible space by adding $m - 1$ hyperplanes through a selected point $\bar{\mathbf{J}}_p$ in the utopia plane. These hyperplanes are chosen perpendicular to each of the $m - 1$ *utopia plane vectors*, which join the (normalised) individual minimum $\bar{\mathbf{J}}(\mathbf{y}_k^*)$ corresponding to the selected objective $\bar{J}_k$, with all other (normalised) individual minima $\bar{\mathbf{J}}(\mathbf{y}_i^*)$. Hence, this ap-

proach leads to additional inequality constraints:

$$\min_{\mathbf{y} \in \mathcal{S}} \bar{\mathbf{J}}_k \text{ s.t. : } (\bar{\mathbf{J}}(\mathbf{y}_k^*) - \bar{\mathbf{J}}(\mathbf{y}_i^*))^T (\bar{\mathbf{J}}(\mathbf{y}) - \bar{\mathbf{J}}_p) \leq 0$$

$$i = 1 \dots m, i \neq k. \quad (9)$$

As in NBI, evenly distributed points on the utopia plane $\bar{\mathbf{J}}_p$ can be selected by a uniform variation of a ‘weight’ vector $\mathbf{w} = [w_1, w_2, \dots, w_i, \dots, w_m]^T$ (with $\sum_{i=1}^m w_i = 1$ and $w_i \geq 0$), which also ensures an even spread on the Pareto set. In addition, an enhanced normalisation scheme by Sanchis et al (2008) for cases with $m \geq 3$ can be used.

Remark. In the case of highly non-convex and discontinuous Pareto sets, both NBI and NNC can generate non-Pareto optimal points. The reasons can be related to (i) the scalarisation approach itself (e.g., when the quasi-normal in NBI intersects the boundary of the feasible region in a region which is not Pareto optimal) or (ii) the local convergence of fast gradient based optimisation algorithms (e.g., when in NNC a local and not a global minimum is found for the minimisation of the selected objective J_k). (For more details see the original papers by Das and Dennis (1998) and Messac et al (2003)). However, it should be emphasised that these unfortunate cases are not frequently encountered in real-world engineering problems. Moreover, the candidate solution set can be filtered afterwards using a Pareto filter algorithm (see, e.g., Messac et al (2003)), to remove non-Pareto optimal points.

3.2 Direct approaches for optimal control problems

Numerical methods for solving optimal control problems are generally classified in *direct* and *indirect* approaches (e.g., Srinivasan et al (2003)). Methods of the latter class try to find solutions according to Pontryagin’s Minimum Principle (Pontryagin et al 1962), i.e., the first-order necessary conditions for optimality, while techniques of the former class convert the original infinite dimensional optimal control problem into a finite dimensional Non-Linear Programming problem (NLP) via discretisation. However, most optimal control problems are now solved by direct approaches.

3.2.1 Sequential approaches: Single Shooting

In Single Shooting (SiS) (Sargent and Sullivan 1978; Vassiliadis et al 1994a,b) only the controls $\mathbf{u}$ are discretised, most often using piecewise polynomials. For each parameterisation the differential equations are solved using a standard integration algorithm, and the objective function is evaluated. The parameter values are

then updated employing a standard optimisation algorithm. Hereto, most often a deterministic, Sequential Quadratic Programming (SQP) routine is employed. Consequently, the solution of the differential equations and the minimisation are decoupled, and proceed sequentially within one iteration step of the optimiser. As these methods provide a correct solution to the differential equations during all iterations, they are also known as *feasible path* methods.

The most important advantage of the sequential approach is its straightforward implementation resulting in rather small-scale NLPs. However, the sequential approach may be slow, especially in the presence of path constraints on the states, since these constraints cannot be enforced directly.

3.2.2 Simultaneous approaches: Multiple Shooting and Orthogonal Collocation

Simultaneous approaches discretise both the control and the states. Consequently, the simulation and optimisation are performed simultaneously in the space of both the discretised controls and states, yielding a large-scale NLP which requires tailored numerical optimisation methods. Since in this case the differential equations are only satisfied at the solution of the optimisation problem, these methods are also called *infeasible path* methods.

Two different approaches exist: Multiple Shooting (MuS) (Bock and Plitt 1984; Leineweber et al 2003b) and Orthogonal Collocation (OCol) (Biegler 1984, 2007). In the former case, the total integration range is split into a finite number of intervals on which integration of the states is continuous. The value of the control and the initial value of the states in each interval are chosen by the NLP-solver in each iteration while trying to ensure the continuity of the states between the different intervals by adding additional equality constraints. In the latter case, also the states are fully discretised based on (orthogonal) polynomials. Hence, the minimal objective value has to be found while satisfying the discretised differential equations. Clearly, MuS and OCol allow a more direct enforcement of the state constraints. However, the size of the NLPs significantly increases, requiring tailored optimisation algorithms that exploit the problem’s structure and sparsity. Obviously, the largest but also sparsest NLPs are encountered with OCol. Consequently, most often Interior Point (Wächter 2002) and partially reduced SQP methods (Leineweber et al 2003a) are employed for OCol and MuS, respectively.

Fig. 2 Schematic outline of the ACADO Multi-Objective toolkit (www.acadotoolkit.org).

4 Implementation into ACADO Multi-Objective

This section describes the software package **ACADO Multi-Objective** (see also Figure 2), which is the multiple objective extension of the **ACADO** toolkit. **ACADO** is a freely available stand-alone toolkit for automatic control and dynamic optimisation (Houska et al accepted). The rationale behind **ACADO Multi-Objective** is the integration of accurate scalarisation techniques with fast deterministic direct methods for optimal control. Weights are generated automatically and several re-initialisation strategies for solving the different parametric single objective optimal control problems have been incorporated in order to speed up the convergence and to reduce the CPU time. A Pareto filter algorithm has been added in order to remove possible non-Pareto optimal points. Finally different tools for visualising and exporting the results have been included.

Because of its self-contained, object-oriented C++ source code, the toolkit is generic, easy-to-use and flexibly adaptable. To solve an MOOCP the user only has to provide a single file (e.g., `MyMOOCP.cpp`) with the problem formulation, in which also the different algorithmic options can be set in the following format:

```
Algorithm.set("OptionName", "OptionValue").
```

When no specific values are provided by the user, default values are adopted. Compilation of the problem file ensures a fast execution and low CPU times.

The most important features for solving MOOCPs are summarised in the following list:

- **Multiple objective optimisation methods.** Three deterministic scalarisation methods have been implemented: WS, NBI and NNC (without and with the enhanced normalisation scheme by Sanchis et al (2008)). The implementation is generic such that problems with any number of objectives can be tackled. The additional constraints resulting from the NBI and NNC reformulations are dealt with by the NLP-solver.
- **Weight generation.** The weights are generated automatically based on a given step size. Currently, only convex combinations can be generated, i.e., satisfying $\sum_{i=1}^m w_i = 1$ and $w_i \geq 0$. However, more advanced weight generation schemes, which leave out the positivity constraints (see, e.g., Messac and Mattson (2004)), can be incorporated as well. These negative weights are helpful to avoid the overlooking of Pareto points by NBI and NNC in cases with more than two objectives.
- **Single objective optimisation problem initialisation.** Different initialisation strategies for the series of SOOPs can be selected. First, all SOOPs can be initialised using the same fixed values provided by the user. Second, a hot-start strategy, which exploits the result of the previous SOOP to initialise the next one, most often allow a significant decrease in computation time. Finally, also the weights can be used to generate an initial guess based on a linear combination of the minimisers of the individual objectives.
- **Direct optimal control methods.** In the current version of **ACADO**, only SiS and MuS are available. However, it is planned that OCol approaches will be implemented in the near future.
- **Integration methods.** Different integration routines are available for ordinary differential equation systems, e.g., explicit integrators as Runge-Kutta45 or Runge-Kutta78 and implicit integrators as backward differentiation formulae. It should be noted that this last type can also be exploited to solve systems of differential algebraic equations. In addition, all have been equipped with forward and backward automatic differentiation options. Settings that can be adapted by the user involve, e.g., the absolute and relative integration tolerances.
- **Optimisation routines.** As underlying optimisation routines for solving the NLPs, SQP and Interior Point type methods are available. Optimisation settings to be specified relate to, e.g., the Karush-Kuhn-Tucker optimisation tolerance.
- **Visualisation.** The resulting Pareto sets can be directly plotted for cases with up to 3 objectives. The optimal state and control profiles and their cor-

responding Pareto optimal objective values can be exported in different formats, e.g., `.txt` files, which can be imported into graphical software or `.mat` files which can be directly imported into `Matlab`.

- **Pareto filter.** As NBI and NNC may produce non-Pareto optimal points, the generated solution set can be filtered using a Pareto filter algorithm. This filter simply works by comparing a generated point with every other generated point, and if a point is not Pareto optimal, it is eliminated.

5 Examples

In this section several examples are presented in order to illustrate the `ACADO Multi-Objective` toolkit. First, two static examples are tackled, while afterwards four optimal control cases from different engineering disciplines are discussed. As the objectives in all cases are contradicting, NBI and NNC have been employed to generate the results. Although available, no explicit results for WS are given, since similar advantages for NBI and NNC have been found as the ones mentioned in previous comparative studies (see, e.g., Das and Dennis (1998); Messac et al (2003) for static and Logist et al (2009)) for optimal control cases). As NBI and NNC often yield identical results, only the NNC results are presented. However, remarks are added whenever differences have been observed. Finally, also an overview of the computational details is presented for all cases.

5.1 Static optimisation with two objectives

As a first static example, the bi-objective case studied by Messac et al (2003) is selected. It concerns the simultaneous minimisation of $J_1 = y_1$ and $J_2 = y_2$ subject to the bounds:

$$0 \leq y_1 \leq 5.0 \quad (10)$$

$$0 \leq y_2 \leq 5.2 \quad (11)$$

and the non-linear inequality constraint:

$$y_2 \geq 5 \exp(-y_1) + 2 \exp(-0.5(y_1 - 3)^2). \quad (12)$$

The results are depicted in Figure 3. Although the Pareto set is clearly non-convex both NBI and NNC are able to generate solutions with a nice spread along the entire front. Due to the stringent additional equality constraints, NBI returns several non-Pareto optimal points, while NNC is able to avoid some of these solutions because of its less restrictive additional inequality constraints. Nevertheless, the produced non-Pareto optimal points are easily removed by the Pareto filter algorithm.

Fig. 3 Static bi-objective example: Pareto front with 41 points obtained with NBI and NNC, with and without application of the Pareto filter.

5.2 Static optimisation with three objectives

To illustrate the general nature of the toolkit's implementation, the second example deals with a static tri-objective optimisation problem. The aim is to simultaneously minimise $J_1 = y_1$, $J_2 = y_2$ and $J_3 = y_3$ given bounds and a non-linear inequality constraint:

$$-5 \leq y_1, y_2, y_3 \leq 5 \quad (13)$$

$$y_1^2 + y_2^2 + y_3^2 - 4 \leq 0. \quad (14)$$

Figure 4 displays the obtained results, illustrating the toolkit's general implementation that allows to solve in principle MOOPs with any number of objective functions. Clearly, also in this case a nice spread is observed. However, it should be noted that Pareto optimal points near the boundary have been overlooked (see, also Das and Dennis (1998); Messac and Mattson (2004)). Nevertheless, more advanced weight generation strategies (see, e.g., Messac and Mattson (2004)), which remove the positivity requirement for the individual weights, can be exploited to overcome this drawback.

Fig. 4 Static tri-objective example: Pareto front with 66 points obtained with NNC.

Fig. 5 Car example: Pareto front with 11 points obtained with NNC.

5.3 Car

The first MOOCP concerns the transfer of a car from an initial position to a specified target in minimum time and with a minimum fuel consumption (see, e.g., Van Erdeghem et al (2008)). A simple dynamic model for the car is:

$$\frac{dx_1}{dt} = x_2 \quad (15)$$

$$\frac{dx_2}{dt} = u \quad (16)$$

where the time t [s] is the independent variable, the position x_1 [m], and the velocity x_2 [m/s] are the states, and the control u is the acceleration [m/s²], which is in practice controlled by pushing the accelerator, or hitting the brakes. The aim is to drive 400 m, starting and ending at rest:

$$x_1(0) = 0 \text{ m} \quad \text{and} \quad x_1(t_f) = 400 \text{ m} \quad (17)$$

$$x_2(0) = 0 \text{ m/s} \quad \text{and} \quad x_2(t_f) = 0 \text{ m/s} \quad (18)$$

while minimising on the one hand, the control effort for accelerating (which can be interpreted as the fuel consumption):

$$J_1 = \int_0^{t_f} \max(0, u) dt \quad (19)$$

and on the other hand the (scaled) travelling time:

$$J_2 = \frac{t_f}{K} \quad (20)$$

with $K = 20$. Note that the first objective can smoothly be reformulated by using the problem's symmetry. Objectives J_1 and J_2 are obviously conflicting objectives since a small travelling time requires a high speed, and, hence, also a large consumption of fuel for reaching this velocity. Since infinitely fast accelerating and decelerating is impossible, the control is bounded between:

$$-5 \text{ m/s}^2 \leq u(t) \leq 5 \text{ m/s}^2. \quad (21)$$

Fig. 6 Car example: optimal controls obtained with NNC.

Additional constraints are speed and time limits:

$$x_2(t) \leq 40 \text{ m/s} \quad (22)$$

$$t_f \leq 100 \text{ s}. \quad (23)$$

The resulting Pareto frontiers from the NNC approach are depicted in Figure 5. As expected a clear trade-off between time and fuel consumption is visible.

The optimal controls are displayed in Figure 6 and the different phases can be intuitively explained. First, accelerating as fast as possible, then cruising at a certain speed, and finally hitting the brake as hard as possible in order to arrive at rest at the destination. Emphasising one of the objectives is achieved by selecting the scalarisation parameters $1 - w$ and w . Here, w -values close to 0 focus on minimising the first objective, whereas w -values close to 1 concentrate on the minimisation of the second objective. When more emphasis is put on minimising the time (i.e., w close to 1), the acceleration and braking phases last longer (leading to shorter cruising periods, but at a higher speed). On the other hand, emphasising a minimal fuel consumption (i.e., w close to 0) induces short acceleration and braking periods and long cruising phases (at lower speeds).

5.4 Home heating system

The second MOOCP case study involves the optimal control of a small heat pump coupled to a floor heating system in view of conflicting energy and thermal comfort objectives. A dynamic model has been reported by Bianchi (2006):

$$\frac{dx_1}{dt} = \frac{-k_{WR}}{\rho_W c_{pW} V_H} x_1 + \frac{k_{WR}}{\rho_W c_{pW} V_H} x_2 + \frac{u}{\rho_W c_{pW} V_H} \quad (24)$$

$$\frac{dx_2}{dt} = \frac{k_{WR}}{k_G \tau_G} x_1 - \frac{k_{WR} + k_G}{k_G \tau_G} x_2 + \frac{d}{\tau_G} \quad (25)$$

Fig. 7 Home heating example: Pareto front with 11 points obtained with NNC.

where the time t [s] is the independent variable. The states x_1 [°C] and x_2 [°C], denote the temperature of the water returning from the heating and the room temperature, respectively. The control u [W] is the heat supplied from the heat pump to the floor, while the outside temperature d [°C] induces a disturbance. Here, ρ_W [kg/m³], c_{pW} [J/kgK] and V_H [m³] are the density, the specific heat and the volume of the water in the floor heating system. k_{WR} [W/K] and k_G [W/K] are the thermal conductivities between respectively, the water and the room, and the room and the environment, while τ_G [s] indicates the thermal time constant of the room.

Initially, the water and room temperature are both at 19.5°C. The objective is to minimise the (normalised) energy input over an entire day $t_f=24$ h:

$$J_1 = \int_0^{t_f=24 \text{ h}} \frac{u}{P_{\max}} dt \quad (26)$$

while maximising the thermal comfort, i.e., remaining as close a possible to a desired value $x_{2,\text{ref}}=20^\circ\text{C}$:

$$J_2 = \int_0^{t_f=24 \text{ h}} (x_2 - x_{2,\text{ref}})^2 dt \quad (27)$$

despite a sinusoidal disturbance originating from the outside temperature:

$$d(t) = 2.5 + 7.5 \sin(2\pi t/t_f - \pi/2). \quad (28)$$

However, due to constructive reasons, the heating power of the heat pump is limited.

$$0 \text{ W} \leq u(t) \leq P_{\max} = 15000 \text{ W} \quad (29)$$

The resulting trade-off between thermal comfort and energy objectives is depicted in Figure 7. At the one extreme, it is seen that adding no energy at all results in a high comfort objective (and thus a low comfort). Increasing the energy input causes a gradual decrease in

Fig. 8 Home heating example: optimal controls obtained with NNC.

the thermal comfort objective (i.e., improves the comfort). However, when approaching the other extreme, it is seen that much energy has to be invested for only a (marginal) gain in thermal comfort. Hence, the current Pareto chart can help control engineers in practice to decide on an appropriate operation policy.

The increase in energy effort is also seen in the control profiles (see Figure 8). Again, w -values close to 0 indicate a focus on the first objective, i.e., minimising energy input, while cases with w -values close to 1 concentrate on the second objective, i.e., maximising the thermal comfort. A general heating profile consists of a phase of heating at full capacity in order to counteract the cold of the night, while afterwards it is decreased since during the day the ambient temperature is higher and less heat is lost.

5.5 Fed-batch bioreactor

The third MOOCP studies the optimal feeding rate and batch duration for a fed-batch lysine fermentation process in view of conflicting yield and productivity objectives (Logist et al 2009). Assuming perfect mixing and no product degradation yields the following model equations:

$$\frac{dx_1}{dt} = +\mu x_1 \quad (30)$$

$$\frac{dx_2}{dt} = -\sigma x_1 + u C_{s,F} \quad (31)$$

$$\frac{dx_3}{dt} = +\pi x_1 \quad (32)$$

$$\frac{dx_4}{dt} = +u \quad (33)$$

with the time t [h] as the independent variable. The states variables are x_1 [g], the biomass, x_2 [g], the substrate, x_3 [g], the product (lysine), and x_4 [L] the fer-

Fig. 9 Bioreactor example: Pareto front with 41 points obtained with NNC.

menter volume. The control u [L/h] is the volumetric rate of the feed stream, which contains a limiting substrate concentration $C_{s,F}$ of 2.8 g/L. The specific rates for growth μ [1/h], substrate consumption σ [g/gh], and production π [g/gh] are:

$$\mu = 0.125C_s \quad (34)$$

$$\sigma = \mu/0.135 \quad (35)$$

$$\pi = -384\mu^2 + 134\mu \quad (36)$$

with C_s [g/L] the substrate concentration. The initial conditions are specified as:

$$x_1(0) = 0.1 \text{ g} \quad (37)$$

$$x_2(0) = 14 \text{ g} \quad (38)$$

$$x_3(0) = 0 \text{ g} \quad (39)$$

$$x_4(0) = 5 \text{ L}. \quad (40)$$

The aim is to derive a feeding strategy and batch duration which maximise the productivity, i.e., the ratio between the product formed and the process duration:

$$J_p = \frac{x_3(t_f)}{t_f} \quad (41)$$

while maximising the yield, i.e., the mass of product over the mass of substrate added during the operation:

$$J_y = \frac{x_3(t_f)}{(x_4(t_f) - x_4(0))C_{s,F}}. \quad (42)$$

To cast these maximisation problems into a minimisation framework the objective functions are defined as the negative productivity and yield: $J_1 = -J_p$ and $J_2 = -J_y$. For practical reasons constraints are imposed on the volume, the feed rate, the operation time

Fig. 10 Bioreactor example: optimal controls obtained with NNC.

and the amount to be added.

$$5 \text{ L} \leq x_4(t) \leq 20 \text{ L} \quad (43)$$

$$0 \text{ L/h} \leq u(t) \leq 2 \text{ L/h} \quad (44)$$

$$20 \text{ h} \leq t_f \leq 40 \text{ h} \quad (45)$$

$$20 \text{ g} \leq (x_4(t_f) - x_4(0))C_{s,F} \leq 42 \text{ g} \quad (46)$$

Figure 9 depicts the Pareto frontier as obtained with NBI or NNC. Clearly, the objectives are not only conflicting but there also exists a non-convex region in the Pareto front. Both NBI and NNC are able to cope with this difficulty.

The optimal feeding profiles consist of arc sequences of maximum feeding, followed by minimum, intermediate (so-called *singular*) and again minimum feeding (see Figure 10). When only productivity is aimed at (i.e., $w = 0$), the initial *max* part is present in order to stimulate biomass growth, and hence lysine production. However, when the focus shifts towards yield optimisation (i.e., w going towards 1), the *max* piece shortens (and vanishes) and the singular profile reduces in height but lasts longer. In general, the total process duration increases towards the upper limit of 40 h allowing the production of more lysine with the same amount of substrate (but at the expense of longer process times). However, there exist a region (around w -values of 0.75) where the optimal duration decreases as the gain in productivity by shorter fermentation times apparently dominates the loss in yield.

5.6 Crane

As a final MOOCP case, the robust control of a 2D crane is considered. Here, the crane model, which is taken from Houska and Diehl (2009), consists of a trolley, a mass point, and a cable connecting the trolley

with the mass. There are two control inputs: first the acceleration a_y [m/s²] can be used to influence the horizontal position y [m] as well as the associated velocity $\dot{y}$ [m/s] of the trolley. Second, the acceleration a_L [m/s²] of the cable can be used to influence the cable length L [m] and its elongation speed $\dot{L}$ [m/s]. The excitation of the mass point can be described by a pendulum model for the nominal excitation angle ϕ_n [rad] and the associated nominal angular velocity $\dot{\phi}_n$ [rad/s]. The dynamic equations can be summarised as follows:

$$\frac{dy}{dt} = \dot{y} \quad (47)$$

$$\frac{d\dot{y}}{dt} = a_y \quad (48)$$

$$\frac{dL}{dt} = \dot{L} \quad (49)$$

$$\frac{d\dot{L}}{dt} = a_L \quad (50)$$

$$\frac{d\phi_n}{dt} = \dot{\phi}_n \quad (51)$$

$$\frac{d\dot{\phi}_n}{dt} = -\frac{g}{L}\phi_n - \left(b + 2\frac{\dot{L}}{L}\right)\dot{\phi}_n - \frac{a_y}{L} \quad (52)$$

where $g = 9.81$ m/s² denotes the gravitational constant and $b = 0.2$ 1/s is a damping coefficient.

The interesting aspect about the current case study is that the crane is subject to uncertainty (due to, e.g., wind). It has been observed that the mass of the crane is affected by a random force F , which is assumed to be a stochastic white noise disturbance. This leads to the following equations:

$$\frac{d\phi}{dt} = \frac{d\phi_n}{dt} \quad (53)$$

$$\frac{d\dot{\phi}}{dt} = \frac{d\dot{\phi}_n}{dt} + \frac{F}{mL} \quad (54)$$

with

$$\begin{aligned} \forall t \in [0, t_f] : \quad \mathbb{E}\{F(t)\} &= 0 \\ \forall t, t' \in [0, t_f] : \quad \mathbb{E}\{F(t)F(t')\} &= \Sigma_F \delta(t - t'), \end{aligned} \quad (55)$$

where δ denotes the Dirac distribution. In this example, $\Sigma_F = 50$ N²/s is chosen, with a mass m of 10 kg.

Hence, not only information about the nominal expectation values $\phi_n = \mathbb{E}\{\phi(t)\}$ and $\dot{\phi}_n = \mathbb{E}\{\dot{\phi}(t)\}$ is available, but also uncertainty information, which is propagated through the dynamic equations for the variance-covariance information $P_{\phi,\phi}(t) = \mathbb{E}\{\phi(t)^2\}$, $P_{\phi,\dot{\phi}}(t) = \mathbb{E}\{\phi(t)\dot{\phi}(t)\}$, and $P_{\dot{\phi},\dot{\phi}}(t) = \mathbb{E}\{\dot{\phi}(t)^2\}$:

$$\frac{dP_{\phi,\phi}}{dt} = 2P_{\phi,\dot{\phi}} \quad (56)$$

$$\frac{dP_{\phi,\dot{\phi}}}{dt} = -\frac{g}{L}P_{\phi,\phi} - \left(b + 2\frac{\dot{L}}{L}\right)P_{\phi,\dot{\phi}} + P_{\dot{\phi},\dot{\phi}} \quad (57)$$

$$\frac{dP_{\dot{\phi},\dot{\phi}}}{dt} = -2\frac{g}{L}P_{\phi,\dot{\phi}} - 2\left(b + 2\frac{\dot{L}}{L}\right)P_{\dot{\phi},\dot{\phi}} + \frac{\Sigma_F}{m^2 L^2} \quad (58)$$

The boundary conditions:

$$y(0) = 0 \text{ m} \quad \text{and} \quad y(t_f) = 10 \text{ m} \quad (59)$$

$$\dot{y}(0) = 0 \text{ m/s} \quad \text{and} \quad \dot{y}(t_f) = 0 \text{ m/s} \quad (60)$$

$$L(0) = 70 \text{ m} \quad \text{and} \quad L(t_f) = 70 \text{ m} \quad (61)$$

$$\dot{L}(0) = 0 \text{ m/s} \quad \text{and} \quad \dot{L}(t_f) = 0 \text{ m/s} \quad (62)$$

$$\phi_n(0) = 0 \text{ rad} \quad (63)$$

$$\dot{\phi}_n(0) = 0 \text{ rad/s} \quad (64)$$

$$P_{\phi,\phi}(0) = 0 \quad (65)$$

$$P_{\phi,\dot{\phi}}(0) = 0 \quad (66)$$

$$P_{\dot{\phi},\dot{\phi}}(0) = 0 \quad (67)$$

guarantee that the crane is started completely at rest at the position $y(0) = 0$ m and with an initial cable length of $L(0) = 70$ m. Note that the initial values are assumed to be exactly known since the variance-covariance states are initialised with 0.

The path constraints take the following form:

$$-0.3 \text{ m/s}^2 \leq a_y(t) \leq 0.3 \text{ m/s}^2 \quad (68)$$

$$-1.0 \text{ m/s}^2 \leq a_L(t) \leq 1.0 \text{ m/s}^2 \quad (69)$$

$$-1.0 \text{ m} \leq y(t) \leq 11.0 \text{ m} \quad (70)$$

$$-20.0 \text{ m/s} \leq \dot{y}(t) \leq 20.0 \text{ m/s} \quad (71)$$

$$30.0 \text{ m} \leq L(t) \leq 75.0 \text{ m} \quad (72)$$

$$-20.0 \text{ m/s} \leq \dot{L}(t) \leq 20.0 \text{ m/s} \quad (73)$$

$$-0.15 \text{ rad} + \gamma\sqrt{P_{\phi,\phi}} \leq \phi_n(t) \leq 0.15 \text{ rad} - \gamma\sqrt{P_{\phi,\phi}} \quad (74)$$

$$-0.1 \frac{\text{rad}}{\text{s}} + \gamma\sqrt{P_{\dot{\phi},\dot{\phi}}} \leq \dot{\phi}_n(t) \leq 0.1 \frac{\text{rad}}{\text{s}} - \gamma\sqrt{P_{\dot{\phi},\dot{\phi}}} \quad (75)$$

with γ a positive confidence level which controls the single chance probability of a constraint violation (Houska and Diehl 2009). The crane is required to move the mass into a given target region described by the following terminal inequality constraints:

$$\begin{aligned} -0.075 \text{ rad} + \gamma\sqrt{P_{\phi,\phi}(t_f)} &\leq \phi_n(t_f) \\ &\leq 0.075 \text{ rad} - \gamma\sqrt{P_{\phi,\phi}(t_f)}. \end{aligned} \quad (76)$$

There are two conflicting interests: on the one hand, the crane has to bring the mass m as fast as possible into the target region, while satisfying all constraints:

$$J_1 = t_f. \quad (77)$$

Fig. 11 Crane example: Pareto front with 31 points obtained with NNC.

On the other hand, the constraints have to be robustly satisfied. Thus, the single chance probability $p(\gamma)$ of a constraint violation has to be minimised² too:

$$J_2 = \frac{1}{\sqrt{2\pi}} \int_{\gamma}^{\infty} e^{-\frac{\tau^2}{2}} d\tau = \frac{1}{2} \left(1 - \operatorname{erf} \left(\frac{\gamma}{\sqrt{2}} \right) \right). \quad (78)$$

In Figure 11 the resulting Pareto front is depicted. Note that there is a clear non-convex trade-off between the two objectives. If the crane performs the manoeuvre in the fastest time (i.e., 11.55 s) the single chance probability of a constraint violation is about 23%. On the other hand, if the crane manoeuvre is optimised with respect to robustness, the lowest constraint violation probability $p(\gamma_{\min}) = 6.8\%$ is achieved. Hence, there exists an intrinsic and substantial probability that the constraints cannot be met. The price to be paid for the lower probability is that the manoeuvre takes in this case $t_f = 16.95$ s. The increase in duration is also visible in Figure 12, where the corresponding controls are depicted. In the current case, w -values close to 0 indicate control strategies for fast manoeuvring, while slower but more robust control strategies are given for w -values close to 1.

Note that this kind of discussion is typical for robust optimisation problems. In fact, robustness or safety typically comes along with a higher investment cost in terms of the nominal objective. In practice, the final decision on how robust an implementation must be can often not only be made by optimisation software, but must be taken in cooperation with experienced engineers. A fast and systematic generation of equally valid

² Note that the objective J_2 can also be replaced by $-\gamma$, because a maximisation of γ is equivalent to a minimisation of the constraint violation probability. However, the latter has a more intuitive interpretation.

Fig. 12 Crane example: optimal controls obtained with NNC.

solutions (i.e., sets of Pareto optimal points) for robustness optimisation problems supports and enhances decision making in practice. Hence, the presented tool also allows engineers in practice to deal swiftly with robustness trade-offs.

5.7 Discussion

Table 1 presents for the different MOOCP cases an overview of the computational burden as measured on a 1.86 GHz computer with 2 GB RAM. The total number of SQP iterations and the total CPU time required to generate the entire Pareto front are given, as well as the average CPU time per Pareto point. With respect to the different scalarisation methods, NBI and NNC induce a similar computational burden, which is higher than the one for WS, due to the additional equality and inequality constraints. However, it should be taken into account that the Pareto sets generated by WS in general do not achieve the same accuracy. Pareto points are missed, e.g., in non-convex parts of the Pareto set or due to bad objective scaling and low sensitivity. For instance, WS cannot provide a descent approximation of the non-convex Pareto front in the crane example. The average computation times per Pareto point vary between 0.2 and 2 s. As expected, the cases with singular arcs (e.g., the bioreactor example), require more CPU time to converge than the cases with only *max*, *min* and *constrained* arcs (e.g., the crane example), since the sensitivity with respect to the controls is in these cases typically lower. Also the hot-start strategy speeds up the convergence by a factor of 2 to 3, roughly speaking. With respect to the direct optimal control approaches, no real differences between MuS and SiS are observed as none of the current engineering examples exhibits unstable dynamics or a high number of active constraints.

Table 1 Overview of computational results for the different MOOCP cases. (NS: No correct solution returned.)

	Car			Heating			Bioreactor			Crane		
	WS	NBI	NNC	WS	NBI	NNC	WS	NBI	NNC	WS	NBI	NNC
# states $\mathbf{x}(t)$	2	2	2	2	2	2	4	4	4	9	9	9
# controls $\mathbf{u}(t)$	1	1	1	1	1	1	1	1	1	2	2	2
End time t_f	free	free	free	fixed	fixed	fixed	free	free	free	free	free	free
# discretisation intervals	50	50	50	48	48	48	50	50	50	20	20	20
# Pareto points	11	11	11	11	11	11	41	41	41	31	31	31
MuS: SQP iterations	103	190	190	195	261	259	636	512	487	NS	167	195
total CPU time [s] (front)	2.10	3.72	3.73	13.00	17.12	17.14	88.51	68.29	68.79	NS	15.02	17.71
CPU time per Pareto point [s]	0.19	0.34	0.34	1.18	1.56	1.56	2.16	1.67	1.68	NS	0.48	0.57
CPU speed-up factor	5.2	1.5	2.5	1.8	2.6	2.9	2.7	2.4	2.5	NS	3.1	4.5
SiS: SQP iterations	124	432	438	190	252	258	544	448	466	NS	165	190
total CPU time [s] (front)	2.56	8.75	8.45	12.81	16.57	17.10	73.35	58.00	59.60	NS	15.81	17.63
CPU time per Pareto point [s]	0.23	0.80	0.77	1.16	1.51	1.55	1.79	1.41	1.45	NS	0.51	0.57
CPU speed-up factor	3.7	1.3	1.3	1.8	2.7	2.8	3.0	2.4	2.7	NS	2.7	2.3

In summary, the **ACADO Multi-Objective** toolkit is able to tackle general non-linear MOOCPs. The cases studied have covered a variety of aspects that can be encountered in practice: (i) linear and non-linear ordinary differential equation systems, (ii) initial and boundary value problems, (iii) (non-linear) path and terminal inequality constraints, (iv) presence and absence of singular arcs, (v) convex and non-convex Pareto sets, etc. Accurate results have been obtained for all cases by using a fine grid for the control discretisation (e.g., with 20 to 50 pieces) and tight tolerances for integration and optimisation (e.g., 10^{-6} and 10^{-7}). Also, it has to be emphasised that no a-priori knowledge about the optimal arcs or the optimal switching sequence is required. Nevertheless, the computation times remain modest. More specifically, the use of hot-start re-initialisation strategy for the different SOOPs causes a significant reduction in the computational effort and ensures a fast solution. Hence, the **ACADO Multi-Objective** can be a valuable tool to support decision-making in real-time as it has been shown to be able to generate a set of equally valid solutions (i.e., Pareto optimal points) within minutes.

6 Conclusions

This paper deals with the fast and efficient solution of optimal control problems with multiple objectives. Hereto, several scalarisation techniques for multi-objective optimisation, e.g., Weighted Sum, Normalised Normal Constraint and Normal Boundary Intersection have been integrated with fast deterministic direct optimal control approaches (e.g., Single Shooting and Multiple Shooting). All techniques have been implemented in **ACADO Multi-Objective**, which is an add-on to the automatic control and dynamic optimisation toolkit

ACADO. Both are freely available at www.acadotoolkit.org. Several advanced options have been incorporated (automatic weights generation, (hot-start) re-initialisation strategies, Pareto filter, ...). In addition, due to its self-contained, object-oriented C++ implementation the toolkit is easy-to-use, does not require third-party software, allows a flexible control over algorithmic settings and can readily be extended. Current developments concern the further implementation of Orthogonal Collocation techniques for optimal control, as well as the integration of different weight generation schemes (Messac and Mattson 2004). Moreover, additional features (Das 1999; Martinez et al 2007) and additional scalarisation techniques as Adaptive Weighted Sum (Kim and de Weck 2005, 2006) can also flexibly be included. In view of readability and understandability, a number of well-documented test examples have been included, which will allow the user to rapidly set-up and solve his/her own MOOCP. Additional information is available from the authors.

Acknowledgements This research was supported by Research Council KUL: CoE EF/05/006 Optimization in Engineering (OP-TEC), OT/09/025/TBA, IOF-SCORES4CHEM KP/09/005, several PhD/postdoc & fellow grants; the Flemish Government via FWO: projects G.0452.04, G.0499.04, G.0211.05, G.0226.06, G.0321.06, G.0302.07, G.0320.08 (convex MPC), G.0558.08 (Robust MHE), G.0557.08, G.0588.09, research community ICCoS and via IWT: McKnow-E, Eureka-Flite+ EU: ERNSI; FP7-HD-MPC (Collaborative Project STREP-grantnr. 223854), Contract Research: Helmholtz Gemeinschaft: viCERP; Austria: ACCM, and the Belgian Federal Science Policy Office: IUAP P6/04 (DYSCO, Dynamical systems, control and optimization, 2007-2011). J. Van Impe holds the chair Safety Engineering sponsored by the Belgian chemistry and life sciences federation essenscia. The scientific responsibility is assumed by its authors. Ir. C. Verhelst and Prof. L. Helsen are gratefully acknowledged for providing information on the home heating case.

List of acronyms

CHIM	Convex Hull of Individual Minima
MOOCP	Multiple Objective Optimal Control Problem
MOOP	Multiple Objective Optimisation Problem
MuS	Multiple Shooting
NBI	Normal Boundary Intersection
NLP	Non-Linear Programming problem
NNC	Normalised Normal Constraint
OCol	Orthogonal Collocation
SOOP	Single Objective Optimisation Problem
SiS	Single Shooting
SQP	Sequential Quadratic Programming
WS	Weighted Sum

References

- Bhaskar V, Gupta S, Ray A (2000) Applications of multi-objective optimization in chemical engineering. *Reviews in Chemical Engineering* 16:1–54
- Bianchi M (2006) Adaptive modellbasierte prädiktive regelung einer kleinwärmepumpen anlage. PhD thesis, ETH, Zurich
- Biegler L (1984) Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. *Computers and Chemical Engineering* 8:243–248
- Biegler L (2007) An overview of simultaneous strategies for dynamic optimization. *Chemical Engineering and Processing: Process Intensification* 46:1043–1053
- Bock H, Plitt K (1984) A multiple shooting algorithm for direct solution of optimal control problems. In: *Proceedings of the 9th IFAC world congress, Budapest*, Pergamon Press
- Das I (1999) On characterizing the ‘knee’ of the Pareto curve based on Normal-Boundary Intersection. *Structural and Multidisciplinary Optimization* 2-3:107–115
- Das I, Dennis J (1997) A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural Optimization* 14:63–69
- Das I, Dennis J (1998) Normal-Boundary Intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization* 8:631–657
- Deb K (2001) *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley, Chichester, London, UK
- Haimes Y, Lasdon L, Wismer D (1971) On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man, and Cybernetics SMC-1*:296–297
- Houska B, Diehl M (2009) Robust nonlinear optimal control of dynamic systems with affine uncertainties. In: *Proceedings of the 48th IEEE Conference on Decision and Control (CDC09)*, Shanghai (China), pp 2274–2279
- Houska B, Ferreau H, Diehl M (accepted) ACADO Toolkit - an open-source framework for automatic control and dynamic optimization. *Optimal Control Applications and Methods*
- Kim I, de Weck O (2005) Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Structural and Multidisciplinary Optimization* 29:149–158
- Kim I, de Weck O (2006) Adaptive weighted sum method for multiobjective optimization: a new method for pareto front generation. *Structural and Multidisciplinary Optimization* 31:105–116
- Leineweber D, Bauer I, Bock H, Schlöder J (2003a) An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization. Part I: theoretical aspects. *Computers and Chemical Engineering* 27:157–166
- Leineweber D, Schäfer A, Bock H, Schlöder J (2003b) An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization. Part II: Software aspects and applications. *Computers and Chemical Engineering* 27:167–174
- Logist F, Van Erdeghem P, Van Impe J (2009) Efficient deterministic multiple objective optimal control of (bio)chemical processes. *Chemical Engineering Science* 64:2527–2538
- Marler R, Arora J (2004) Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization* 26:369–395
- Martinez M, Sanchis J, Blasco X (2007) Global and well-distributed pareto frontier by modified normalized normal constraint methods for bicriterion problems. *Structural and Multidisciplinary Optimization* 34:197–209
- Messac A, Mattson C (2004) Normal constraint method with guarantee of even representation of complete Pareto frontier. *AIAA Journal* 42:2101–2111
- Messac A, Ismail-Yahaya A, Mattson C (2003) The normalized normal constraint method for generating the Pareto frontier. *Structural and Multidisciplinary Optimization* 25:86–98
- Miettinen K (1999) *Nonlinear multiobjective optimization*. Kluwer Academic Publishers, Boston
- Pontryagin L, Boltyanskiy V, Gamkrelidze R, Mishchenko Y (1962) *The mathematical theory of optimal processes*. Wiley - Interscience, New York
- Sanchis J, Martinez M, Blasco X, Salcedo J (2008) A new perspective on multiobjective optimization by enhanced normalized normal constraint method. *Structural and Multidisciplinary Optimization* 36:537–546
- Sargent R, Sullivan G (1978) The development of an efficient optimal control package. In: *Stoer J (ed) Proceedings of the 8th IFIP Conference on Optimization Techniques*, Springer-Verlag, Heidelberg, pp 158–168
- Srinivasan B, Palanki S, Bonvin D (2003) Dynamic optimization of batch processes I. Characterization of the nominal solution. *Computers and Chemical Engineering* 27:1–26
- Van Erdeghem P, Logist F, Smets I, Van Impe J (2008) Improved procedures for multiple objective optimal control problems. In: *Proceedings of the 17th IFAC World Congress*, pp 7802–7807
- Vassiliadis V, Sargent R, Pantelides C (1994a) Solution of a class of multistage dynamic optimization problems. 1. Problems without path constraints. *Industrial and Engineering Chemistry Research* 33:2111–2122
- Vassiliadis V, Sargent R, Pantelides C (1994b) Solution of a class of multistage dynamic optimization problems. 2. Problems with path constraints. *Industrial and Engineering Chemistry Research* 33:2123–2133
- Wächter A (2002) *An interior point algorithm for large-scale nonlinear optimization with applications in process engineering*. PhD thesis, Carnegie Mellon University