Bioprocess Biosyst Eng (2013) 36:151–164
DOI 10.1007/s00449-012-0770-9

ORIGINAL PAPER

Multi-objective optimal control of dynamic bioprocesses using ACADO Toolkit

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

Received: 5 March 2012 / Accepted: 5 June 2012 / Published online: 31 July 2012

Abstract The optimal design and operation of dynamic bioprocesses gives in practice often rise to optimisation problems with multiple and conflicting objectives. As a result typically not a single optimal solution but a set of Pareto optimal solutions exist. From this set of Pareto optimal solutions, one has to be chosen by the decision maker. Hence, efficient approaches are required for a fast and accurate generation of the Pareto set such that the decision maker can easily and systematically evaluate optimal alternatives. In the current paper the multi-objective optimisation of several dynamic bioprocess examples is performed using the freely available ACADO Multi-Objective Toolkit (http://www.acadotoolkit.org). This toolkit integrates efficient multiple objective scalarisation strategies (e.g., Normal Boundary Intersection and (Enhanced) Normalised Normal Constraint) with fast deterministic approaches for dynamic optimisation (e.g., single and multiple shooting). It has been found that the toolkit is able to efficiently and accurately produce the Pareto sets for all bioprocess examples. The resulting Pareto sets are added as supplementary material to this paper.

Keywords Multi-objective optimisation · Dynamic optimisation · Bioprocess

Introduction

Multiple and conflicting objectives appear often in the design and optimisation of dynamic bioprocesses (e.g., [12, 23, 24, 33, 39]). The resulting multi-objective optimisation problems yield a set of so-called Pareto optimal solutions instead of one single optimal solution in single-objective optimisation problems [29]. Once this Pareto set is generated, the decision maker (e.g., the process or design engineer) can select one of the solutions according to his/her own preferences. Hence, the fast and accurate determination of these Pareto sets is of high importance for enhancing real-time decision making in practice.

Broadly speaking two classes of methods for generating the Pareto set exist: *vectorisation* and *scalarisation* methods. Vectorisation methods often involve stochastic evolutionary algorithms [7] and tackle the multi-objective optimisation problem directly. Most often a population of candidate solutions is updated based on repeated cost evaluations to evolve gradually to the Pareto frontier. These methods are often successfully used (see, e.g., [33] for bioreactor case studies). However, these approaches (1) may become time consuming due to the repeated model

Electronic supplementary material The online version of this article (doi:10.1007/s00449-012-0770-9) contains supplementary material, which is available to authorized users.

F. Logist · D. Telen · J. Van Impe (✉)
Department of Chemical Engineering, BioTeC and Optimization in Engineering Center (OPTEC), K.U. Leuven W. de Croylaan 46, 3001 Leuven, Belgium
e-mail: jan.vanimpe@cit.kuleuven.be

F. Logist
e-mail: filip.logist@cit.kuleuven.be

D. Telen
e-mail: dries.telen@cit.kuleuven.be

B. Houska · M. Diehl
Department of Electrical Engineering, SCD and Optimization in Engineering Center (OPTEC), K.U. Leuven Kasteelpark Arenberg 10, 3001 Leuven, Belgium
e-mail: boris.houska@esat.kuleuven.be

M. Diehl
e-mail: moritz.diehl@esat.kuleuven.be

simulations required, (2) are less suited to incorporate constraints exactly, and (3) are limited to rather low-dimensional search spaces. In contrast, scalarisation methods convert the multi-objective optimisation problem into a parametric single-objective optimisation problem [9]. The most popular scalarisation approach is the weighted sum (WS) of the individual objectives. Minimising this WS for different weight values with fast deterministic derivative-based optimisation routines yields an approximation of the Pareto set. However, well-known drawbacks for the WS are that an equal distribution of weights does not necessarily lead to an even spread along the Pareto front, and that points in a non-convex part of the Pareto front cannot be obtained [5]. Recent scalarisation based multiple objective optimisation techniques as normal boundary intersection (NBI) [6] and (enhanced) normalised normal constraint ((E)NNC) [27, 30] are able to mitigate these disadvantages of the WS and still allow the use of fast gradient-based solvers.

Therefore, the rationale is to use NBI and (E)NNC in deterministic direct dynamic optimisation approaches to efficiently solve dynamic bioprocess optimisation problems with multiple objectives. All approaches are implemented in the ACADO Multi-Objective Toolkit [19], which is an extension of the Automatic Control and Dynamic Optimisation Toolkit ACADO [14]. Both are freely available at http://www.acadotoolkit.org.

The paper is structured as follows. “Mathematical formulation” introduces the general mathematical formulation and concepts. “ACADO multi-objective toolkit” describes the ACADO Multi-Objective Toolkit and its features. Results for four bioprocess test examples are presented in “Results”. “Conclusion” summarises the concluding remarks.

Mathematical formulation

In general, dynamic optimisation problems with multiple objectives can be formulated as follows.

$$\min_{\mathbf{x}(t),\mathbf{u}(t),\mathbf{p},t_f} \{J_1,\ldots,J_m\} \tag{1}$$

$$\text{subject to: } \frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}(t),\mathbf{u}(t),\mathbf{p},t) \quad t\in[0,t_f] \tag{2}$$

$$\mathbf{0} = \mathbf{b}_c(\mathbf{x}(0),\mathbf{x}(t_f),\mathbf{p}) \tag{3}$$

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

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

Here, $\mathbf{x}$ is the state variable, 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 $t\in[0,t_f]$) with 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.

$$J_i = h_i(\mathbf{x}(t_f),\mathbf{p},t_f) + \int_0^{t_f} g_i(\mathbf{x}(t),\mathbf{u}(t),\mathbf{p},t)\,dt \tag{6}$$

Whenever needed, the maximisation of a specified objective J'_i is achieved using instead $J_i = -J'_i$ as objective function in the minimisation frame. The admissible set $\mathcal{S}$ is defined as the set of feasible points $\mathbf{y} = (\mathbf{x}(\cdot),\mathbf{u}(\cdot),\mathbf{p},t)$ that satisfy the dynamic equation as well as the boundary, path and terminal constraints.

In multi-objective (MO) optimisation, typically a set of Pareto optimal solutions must be found.

A point $\mathbf{y}_a \in \mathcal{S}$ is Pareto optimal if and only if there is no other point $\mathbf{y}_b \in \mathcal{S}$ with $J_i(\mathbf{y}_b) \leq J_i(\mathbf{y}_a)$ for all $i \in \{1,\ldots,m\}$ and $J_j(\mathbf{y}_b) < J_j(\mathbf{y}_a)$ for at least one $j \in \{1,\ldots,m\}$.

In general terms, a solution is said to be Pareto optimal if there exists no other feasible solution that improves one objective function without worsening another.

ACADO Multi-Objective Toolkit

The ACADO Toolkit is a freely available C++ tool for automatic control and dynamic optimisation [14]. Due to its self-contained nature it does not require third-party software. However, it can also easily be extended or coupled with external packages based on the flexible object-oriented implementation. The syntax which is close to the mathematical problem formulation enhances the user-friendliness. ACADO Multi-Objective Toolkit [19] extends the original ACADO Toolkit with several multi-objective optimisation approaches.

The idea behind ACADO Multi-Objective Toolkit is the efficient combination of scalarisation techniques for multi-objective optimisation with fast deterministic derivative-based direct optimal control methods. In scalarisation methods the original multi-objective optimisation problem is converted into a series of single-objective optimisation problems. Each solution yields one point of the Pareto set. By consistently varying the scalarisation parameter(s) (which are often referred to as weight(s)) an approximation of the Pareto set is obtained. In direct optimal control approaches the original infinite dimensional optimal control problem is transformed via discretisation into a finite dimensional non-linear program (NLP). Sequential strategies (e.g., single shooting [31]) discretise only the controls leading to small but dense NLPs, while simultaneous approaches (e.g., multiple shooting [3] and orthogonal collocation [2]) discretise both the controls and states, resulting in large but structured NLPs. Typically the resulting NLPs are solved by fast deterministic optimisation routines.

Several software packages for the solution of optimal control problems exist. Commercial software is, e.g., gPROMS [25] and PROPT [26]. Non-commercial codes involve, e.g., DynoPC [16], MUSCOD-II [17, 18], DyOS [34] and DOT-cvpSB [13]). However, to the best of the authors' knowledge, none of them offers systematic multi-objective features.

Figure 1 displays the structure of ACADO Multi-Objective Toolkit. The following features have been implemented in the toolkit.

- *Multi-objective optimisation methods*: The implementation exploits scalarisation approaches as WS, NBI, NNC, and ENNC. The ideas behind the methods are the following. NBI first builds a plane in the objective space 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 is that the intersection between the (quasi-)normal from any point on the CHIM, and the boundary of the feasible objective space closest to the utopia point (i.e., the point which contains the minima of the individual objectives) is expected to be Pareto optimal. Hereto, the multi-objective optimisation problem is reformulated as to maximise the distance from a point on the CHIM along the quasi-normal through this point, without violating the original constraints. As a result additional equality constraints are added. (E)NNC exploits similiar ideas but adds inequality constraints representing halfplanes while minimising a selected single objective. As a result $m - 1$ halfplanes are added which are orthogonal to the plane containing all individual minima, now called the utopia plane. The difference between NNC and ENNC is due to a different scaling in the normalisation step. The weights in these methods are typically used to move the points on the CHIM/utopia hyperplane which determines the position of the quasi-normal lines or halfplanes in the criterion space. Hence, a uniform vector results in an even spread of these points on the CHIM/utopia hyperplane and as such it can be understood that a more even spread on the Pareto set can be obtained. The weights in the WS only influence the position in the criterion space in a highly non-linear way [5] and, hence, it can be understood that a uniform spread of weights often does not result in an even spread along the Pareto set. For more info, the interested reader is referred to, e.g., [19].
- *Weight generation*: When a step size for the scalarisation parameter or weight between two of the objectives is specified, a uniformly distributed grid for all the scalarisation parameters or weights w_i is computed automatically. Here, convex weight combinations are generated which satisfy $\sum_{i=1}^{m} w_i = 1$ and $w_i \geq 0$. However, alternative generation schemes that do not require the positivity constraints [28], can be implemented too.
- *Scalability*: In principle any number of objectives can be treated. However, the number of single-objective optimisation problems increases rapidly for increasing number of objectives. When using equally spaced, convex weights (i.e., $w_i \geq 0$, $\sum_{i=1}^{m} w_i = 1$ and n the number of equally spaced points between two individual minima), $(n + m - 2)!/((m - 1)!(n - 1)!)$ single-objective optimisation problems have to be solved. For instance, with three objectives and a stepsize of 0.1, m equals 3 and n equals 11 (i.e. going from 0 to 1 on steps 0.1 yields 11 points). As a result 66 single-objective optimisation problems have to be solved. Hence, for high numbers of objectives, interactive multi-objective methods [29], which interact with the decision maker to explore a preferred region of the objective space, become appealing.
- *Single-objective optimisation problem initialisation*: Different initialisation strategies for the series of single-objective optimisation problems can be selected. All single-objective optimisation problems can be initialised using the same fixed values provided by the user. Alternatively, a hot-start strategy, which exploits the result of the previous single-objective optimisation problem to initialise the next one, allows a significant decrease in computation time. In single shooting only the optimal values for the discretised controls are re-used, while in multiple shooting also the discretised state variabels are re-used. Hence only information about the *primal* variables is exploited and no information about the *dual* variables (*Lagrange multipliers*) is employed.
- *Direct optimal control methods*: ACADO Multi-Objective Toolkit uses the Single and Multiple Shooting methods from the original ACADO Toolkit.
- *Integration routines*: Various integrators are available for ordinary differential equation (ODE) systems. Explicit Runge-Kutta type integrators are RK12 (adaptive Euler), RK23 (second order), RK45 (Dormand-Prince) or RK78 (Dormand-Prince). The BDF integrator is an implicit integrator, which can also tackle systems of (index 1) differential and algebraic equations (DAEs). Integrator settings involve, e.g., the absolute and relative integration tolerances.
- *Sensitivity computation*: The optimal control problem formulation in ACADO Toolkit uses a symbolic syntax which allows storing the functions in the form of

evaluation trees [14]. This enables the use of state of the art automatic differentiation algorithms [10, 11]. In addition as all integrators are equipped with these automatic differentiation features exact first and second-order forward or backward sensitivities of the objective, constraints, and differential equations can be computed with respect to control inputs, parameters, and initial values.

- *Optimisation routines*. Deterministic gradient-based sequential quadratic programming methods (SQP) enable a fast solution of the possibly large-scale NLPs. Although the SQP optimiser can only guarantee local optimality, it has been observed by the authors that chances to get stuck in a local minimum are significantly decreased by exploiting an appropriate initial guess for states and controls in simultaneous approaches such as multiple shooting. Settings to be specified relate to, e.g., the Karush–Kuhn–Tucker optimisation tolerance and the choice between exact and approximate Hessians.
- *Pareto filter*: As NBI and (E)NNC may produce non-Pareto optimal points, the solution set can be filtered using a Pareto filter algorithm [27]. An a posteriori filter simply compares a generated point with every other generated point, and if a point is not Pareto optimal (also called dominated), it is eliminated. In addition, the a priori filter described in Ref. [21] is able to partially remove non-Pareto optimal points without the need for generating a set first. Consequently, a time-saving approach can be to use the a priori filter first and the a posteriori filter afterwards. The first step reduces the number of possible points and, hence, also the number of pairwise comparisons in the second step.
- *Visualisation and output*: The resulting Pareto sets can be directly plotted for cases with up to three objectives. The Pareto optimal cost values and the corresponding optimal profiles for states and controls can be exported in various formats. For higher numbers of objectives the resulting Pareto sets can still be exported but the visualisation becomes more difficult. Bar plots allow a representation but are less clear to interpret. These kinds of plots are not implemented in the toolkit. As mentioned above, the computational complexity becomes large for cases with many objectives, such that interactive methods become interesting.

Remark It has to be noted that NBI and (E)NNC are able to approximate disconneted Pareto sets, as long as the individual minima are accurately found. (See example 1 in Ref. [21].) However, these methods may return non-Pareto optimal points. As mentioned above, these points can be removed by a Pareto filter algorithm.

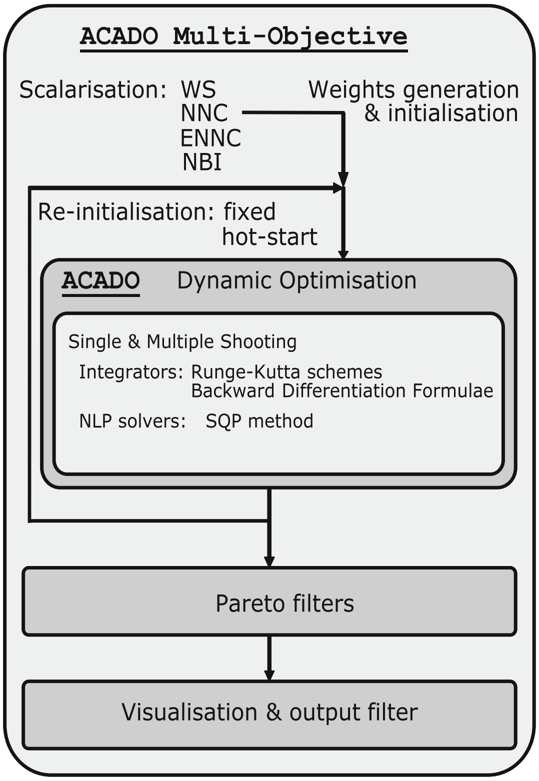
ACADO Multi-Objective
Scalarisation: WS
NNC
ENNC
NBI
Weights generation & initialisation
Re-initialisation: fixed
hot-start
ACADO Dynamic Optimisation
Single & Multiple Shooting
Integrators: Runge-Kutta schemes
Backward Differentiation Formulae
NLP solvers: SQP method
Pareto filters
Visualisation & output filter

Fig. 1 Scheme of the ACADO Multi-Objective Toolkit

Case studies

The approaches are illustrated on four bioprocess case studies, which are detailed in the current section.

Case 1

Case 1 involves the fermentation of glucose to gluconic acid by *Pseudomonas ovalis* in a batch stirred tank reactor as described in Refs. [1, 37, 38].

$$\frac{\mathrm{d}X}{\mathrm{d}t} = \mu_m \frac{SC}{k_s C + k_0 S + SC} X \tag{7}$$

$$\frac{\mathrm{d}p}{\mathrm{d}t} = k_p l \tag{8}$$

$$\frac{\mathrm{d}l}{\mathrm{d}t} = v_l \frac{S}{k_l + S} X - 0.91 k_p l \tag{9}$$

$$\frac{\mathrm{d}S}{\mathrm{d}t} = -\frac{1}{Y_s} \mu_m \frac{SC}{k_s C + k_0 S + SC} X - 1.011 v_l \frac{S}{k_l + S} X \tag{10}$$

$$\frac{\mathrm{d}C}{\mathrm{d}t} = K_L a (C^* - C) - \frac{1}{Y_0} \mu_m \frac{SC}{k_s C + k_0 S + SC} X - 0.09 v_l \frac{S}{k_l + S} X \tag{11}$$

The model consists of the following state variables: X denotes the concentration of cells (UOD/mL), p, the concentration of gluconic acid (g/L), l, the concentration of gluconolactone (g/L), S, the concentration of glucose substrate (g/L) and C, the dissolved oxygen (g/L). The decision variables are the duration of the batch fermentation, $T_B \in [5, 15\text{h}]$, the initial substrate concentration, $S_0 \in [20, 50\text{g/L}]$, the overall oxygen mass transfer coefficient, $K_La \in [50, 3001/\text{h}]$ and the initial biomass concentration, $X_0 \in [0.05, 1.0\text{UOD/mL}]$. The system's initial conditions are given by $[X_0\ 0\ 0\ S_0\ C^*]^T$. Hence, note that only scalar variables have to be optimised. However, two of them are initial conditions. The two objectives are the maximisation of the productivity $J_1 = \frac{p(T_B)}{T_B}$ and the final gluconic acid concentration, $J_2 = p(T_B)$. The parameter values are given in Table 1.

Case 2

The second problem considers the optimal control of a fed-batch reactor for induced foreign protein production by recombinant bacteria as studied by Refs. [1, 32, 38]. The objective is to maximise the profitability of the process using the nutrient and the inducer feeding rates as the control variables. Although this problem was originally solved as a weighted sum between the protein and the inducer cost, Sarkar and Modak [33] have tackled this problem within a systematic multi-objective frame. The dynamic model is the following:

$$\frac{\mathrm{d}x_1}{\mathrm{d}t} = u_1 + u_2 \tag{12}$$

$$\frac{\mathrm{d}x_2}{\mathrm{d}t} = \mu x_2 - (u_1 + u_2)\frac{x_2}{x_1} \tag{13}$$

$$\frac{\mathrm{d}x_3}{\mathrm{d}t} = C_{s,in}\frac{u_1}{x_1} - (u_1 + u_2)\frac{x_3}{x_1} - \mu\frac{x_2}{0.51} \tag{14}$$

$$\frac{\mathrm{d}x_4}{\mathrm{d}t} = \pi x_2 - (u_1 + u_2)\frac{x_4}{x_1} \tag{15}$$

$$\frac{\mathrm{d}x_5}{\mathrm{d}t} = \frac{C_{i,in}u_2}{x_1} - (u_1 + u_2)\frac{x_5}{x_1} \tag{16}$$

$$\frac{\mathrm{d}x_6}{\mathrm{d}t} = -k_1 x_6 \tag{17}$$

$$\frac{\mathrm{d}x_7}{\mathrm{d}t} = k_2(1 - x_7) \tag{18}$$

$$\mu = \frac{x_3}{14.35 + x_3 + \frac{x_3^2}{111.5}}\left(x_6 + x_7\frac{0.22}{0.22 + x_5}\right) \tag{19}$$

$$\pi = \frac{0.233x_3}{14.35 + x_3 + \frac{x_3^2}{111.5}}\frac{0.0005 + x_5}{0.022 + x_5} \tag{20}$$

$$k_1 = \frac{0.09x_5}{0.034 + x_5} \tag{21}$$

$$k_2 = \frac{0.09x_5}{0.034 + x_5} \tag{22}$$

The states are x_1, the reactor volume (L), x_2, the cell density (g/L), x_3, the nutrient concentration (g/L), x_4, the foreign protein concentration (g/L), x_5, the inducer concentration (g/L), x_6 the inducer shock factor on cell growth rate (−) and x_7, the inducer recovery factor on cell growth rate (−). The algebraic states are μ, the specific growth rate; π, the specific foreign protein production rate; k_1, the inducer shock factor; and k_2, the inducer recovery factor. The decision variables are the volumetric rates of the glucose u_1 (L/h) and of the inducer u_2 (L/h). These are bounded between 0 and 1 L/h. The concentrations of inducer and glucose in the feed streams are $C_{i,in} = 4.0$ g/L and $C_{s,in} = 100.0$ g/L, respectively. The initial conditions are $[1\ 0.1\ 40\ 0\ 0\ 1\ 0]^T$ and the final time is fixed at $T_f = 10$ h. As stated above, the objectives are maximising the final amount of foreign protein, $J_1 = x_1(T_f)x_4(T_f)$ and minimising the amount of inducer added, $J_2(T_f) = C_{i,in}\int_0^{T_f} u_2(t)dt$.

Table 1 Case 1: values of the parameters

Parameter	Value	Unit
μ_m	0.39	1/h
k_s	2.50	g/L
k_0	0.00055	g/L
k_p	0.645	1/h
v_l	8.30	mg/UOD h
k_l	12.80	g/L
Y_s	0.375	UOD/mg
Y_0	0.890	UOD/mg
C^*	0.00685	g/L

Case 3

The third case considers a free terminal time fed-batch fermentation process in which ethanol is produced by *Saccharomyces cerevisiae* [4, 22].

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \mu x - u\frac{x}{V} \tag{23}$$

$$\frac{\mathrm{d}s}{\mathrm{d}t} = -x\frac{\mu}{0.1} + u\frac{150 - s}{V} \tag{24}$$

$$\frac{\mathrm{d}p}{\mathrm{d}t} = \eta x - u\frac{p}{V} \tag{25}$$

$$\frac{\mathrm{d}V}{\mathrm{d}t} = u \tag{26}$$

$$\mu = \frac{\mu_0}{1 + \frac{p}{K_p}}\frac{s}{K_s + s} \tag{27}$$

$$\eta = \frac{\eta_0}{1 + \frac{p}{K'_p}} \frac{s}{K'_s + s} \tag{28}$$

The model consists of the following states: x denotes the biomass concentration (g/L), s, the substrate concentration (g/L), p, the product concentration (g/L) and V, the broth volume (L). μ is the specific growth rate (1/h) and η the specific production rate (1/h). The decision variable is the duration of the batch fermentation, $T_f \in [20, 100\mathrm{h}]$ and the time varying feed rate, $u(t) \in [0, 12\mathrm{L/h}]$. An additional constraint implies that the maximal volume is limited to 200 L ($10 \leq V(t) \leq 200$). The initial conditions of the system are specified as $[1\ 150\ 0\ 10]^T$. Here, three different objectives are used. The first objective is maximising productivity $J_1 = \frac{p(T_f)V(T_f)}{T_f}$, the second objective is maximising the production, $J_2 = p(T_f)V(T_f)$ and the third is minimising the substrate cost $J_3(T_f) = \int_0^{T_f} u(t)\,\mathrm{d}t$. However, it has to be ensured that at least 30 L of feed are added, i.e., $\int_0^{T_f} u(t)dt \geq 30$. This constraint can be reformulated as a terminal constraint $x_a(T_f) \geq 30$ on an additional state variable: $\frac{\mathrm{d}x_a}{\mathrm{d}t} = u$ with $x_a(0) = 0$, which measures the amount added. The parameter values are given in Table 2.

Case 4

The fourth and final case involves a fed-batch bioreactor for the production of penicillin G. The model is described in [35, 36]. The difference in model structure with respect to Case 3 is the presence of a biomass constraint and a different inhibition term.

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \mu x - x\frac{u}{V} \tag{29}$$

$$\frac{\mathrm{d}s}{\mathrm{d}t} = -x\frac{\mu}{Y_x} - \frac{v}{Y_p}x + u\frac{s_{in} - s}{V} \tag{30}$$

$$\frac{\mathrm{d}p}{\mathrm{d}t} = vx - p\frac{u}{V} \tag{31}$$

$$\frac{\mathrm{d}V}{\mathrm{d}t} = u \tag{32}$$

$$\mu = \frac{\mu_m s}{K_m + s + (\frac{s^2}{K_i})} \tag{33}$$

Table 2 Case 3: values of the parameters

Parameter	Value	Unit
μ_0	0.408	1/h
K_p	16	g/L
K_s	0.22	g/L
η_0	1.0	1/h
K'_p	71.5	g/L
K'_s	0.44	g/L

As in the previous example the model has four states: x denotes the biomass concentration (g/L), s, the substrate concentration (g/L), p, the product concentration (g/L), and V the broth volume (L). μ is again the specific growth rate (1/h). The batch time T_f is fixed to 150 h. The biomass concentration is limited to 3.7 g/L ($x(t) \leq 3.7$). The decision variable is the time varying feed rate, $u(t) \in [0, 1\mathrm{L/h}]$. The initial conditions are given as $[1\ 0.5\ 0\ 150]^T$. The objectives are to maximise the production, $J_1 = p(T_f)V(T_f)$ and to maximise the concentration of the product or the purity, $J_2 = p(T_f)$ in order to reduce post-processing costs. However to ensure a minimum production, a lower bound of 265 g has been imposed: $J_1 \geq 265$ g. The parameter values are described in Table 3.

Results

This section applies the techniques implemented in the ACADO Multi-Objective Toolkit, i.e., WS, (E)NNC and NBI, to the four dynamic bioprocess optimisation case studies. Each time the Pareto set is displayed as well as several optimal control and state profiles along the Pareto set. The corresponding weight vectors $\mathbf{w} = [w_1, \ldots, w_m]^T$ where the index i runs from 1 to m, are each time indicated. Here w_i relates to the optimisation of objective J_i. Algorithmic settings and the resulting computational expenses in terms of SQP iterations and CPU times are summarised in Table 4. It has to be noted that the model descriptions of cases 2, 3 and 4 include algebraic equations for the sake of clarity. However, the algebraic equations can easily be eliminated by directly substituting them into the differential equations, yielding a system of ordinary differential equations. The objective values in the resulting Pareto sets are added as supplementary material to this paper.

Case 1

As case 1 only involves scalar parameters to be optimised, it is regarded as the simplest of the four cases. The resulting trade-off between maximising productivity and

Table 3 Case 4: values of the parameters

Parameter	Value	Unit
μ_m	0.02	1/h
K_m	0.05	g/L
K_I	5	g/L
Y_x	0.5	(–)
Y_p	1.2	(–)
v	0.004	1/h
S_{in}	200	g/L

Table 4 Overview of algorithmic settings for ACADO Multi-Objective Toolkit and computational expense

	Case 1				Case 2			
	WS	NBI	NNC	ENNC	WS	NBI	NNC	ENNC
# States	5	5	5	4	7	7	7	7
# Controls	0	0	0	0	2	2	2	2
# Dicretisation intervals	25	25	25	25	10	10	10	10
# Pareto points	21	21	21	21	21	21	21	21
Integrator	RK78	RK78	RK78	RK78	BDF	BDF	BDF	BDF
Integrator tolerance	1E−6	1E−6	1E−6	1E−6	1E−6	1E−6	1E−6	1E−6
Hessian	Exact	Exact	Exact	Exact	Exact	Exact	Exact	Exact
Optimality tolerance	1E−6	1E−6	1E−6	1E−6	2E−3	2E−3	2E−3	2E−3
Re-initialisation	Hot-start	Hot-start	Hot-start	Hot-start	Hot-start	Hot-start	Hot-start	Hot-start
SQP iterations	193	290	202	260	139	80	54	76
Total CPU [s] (entire front)	132.4	161.7	118.0	123.0	29.7	19.2	14.1	14.8
Average CPU [s] (1 point)	6.30	7.70	5.62	5.85	1.40	0.92	0.67	0.70

	Case 3				Case 4			
	WS	NBI	NNC	ENNC	WS	NBI	NNC	ENNC
# States	4	4	4	4	4	4	4	4
# Controls	1	1	1	1	1	1	1	1
# Dicretisation intervals	25	25	25	25	20	20	20	20
# Pareto points	66	66	66	66	21	21	21	21
Integrator	BDF	BDF	BDF	BDF	RK78	RK78	RK78	RK78
Integrator tolerance	1E−6	1E−6	1E−6	1E−6	1E−6	1E−6	1E−6	1E−6
Hessian	Exact	Exact	Exact	Exact	Exact	Exact	Exact	Exact
Optimality tolerance	1E−3	1E−3	1E−3	1E−3	1E−4	1E−4	1E−4	1E−4
Re-initialisation	Hot-start	Hot-start	Hot-start	Hot-start	Hot-start	Hot-start	Hot-start	Hot-start
SQP iterations	569	577	421	436	24	56	89	103
Total CPU [s] (entire front)	202.8	140.7	139.1	110.7	1.1	3.0	4.5	5.9
Average CPU [s] (1 point)	3.07	2.13	2.11	1.97	0.052	0.14	0.21	0.28

production is depicted in Fig. 2. Clearly, there is trade-off between maximising production and productivity. It is also seen that the weighted sum does not give an even spread of the Pareto points. In contrast, NBI and (E)NNC return identical results but the distribution of points along the Pareto set is more uniform. The resulting optimal state trajectories obtained with NBI are depicted in Fig. 3. Comparison with results obtained using a global optimisation heuristic in Ref. [37] hardly reveals any differences.

With respect to the scalar optimisation variables, the optimal values for S_0^* and X_0^* are identical to their upper limits. This observation is easily explained as the more biomass and substrate (glucose) is initially present, the faster and higher the production will be. When productivity is focussed on, the highest K_La values and the shortest batch times are encountered. The explanation is that high oxygen transfer rates stimulate biomass growth and early stops avoid a decrease in production rate. However, when

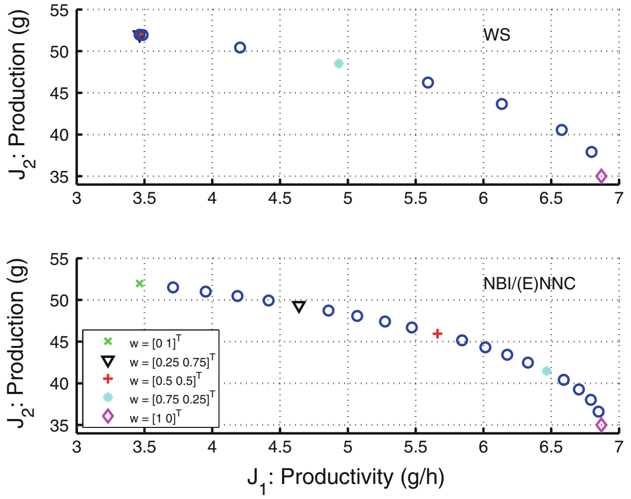

Fig. 2 Case 1: Pareto front with 21 points obtained with WS (*top*) and NBI/(E)NNC (*bottom*)

the total amount of product made in one batch becomes more and more important, longer batch durations and lower K_La values appear to be optimal. In this case, substrate and oxygen are utilised more for the production of gluconic acid, resulting in a slower biomass growth and lower final cell concentrations. Hence, the trade-off depends on how the glucose is used. When focussing on productivity, short batches are preferred and consequently, a fast biomass increase is required. On the other hand, when aiming for production, less substrate is attributed to biomass growth. This results in a slower growth and lower cell numbers, but the glucose is now more directed towards production of gluconic acid.

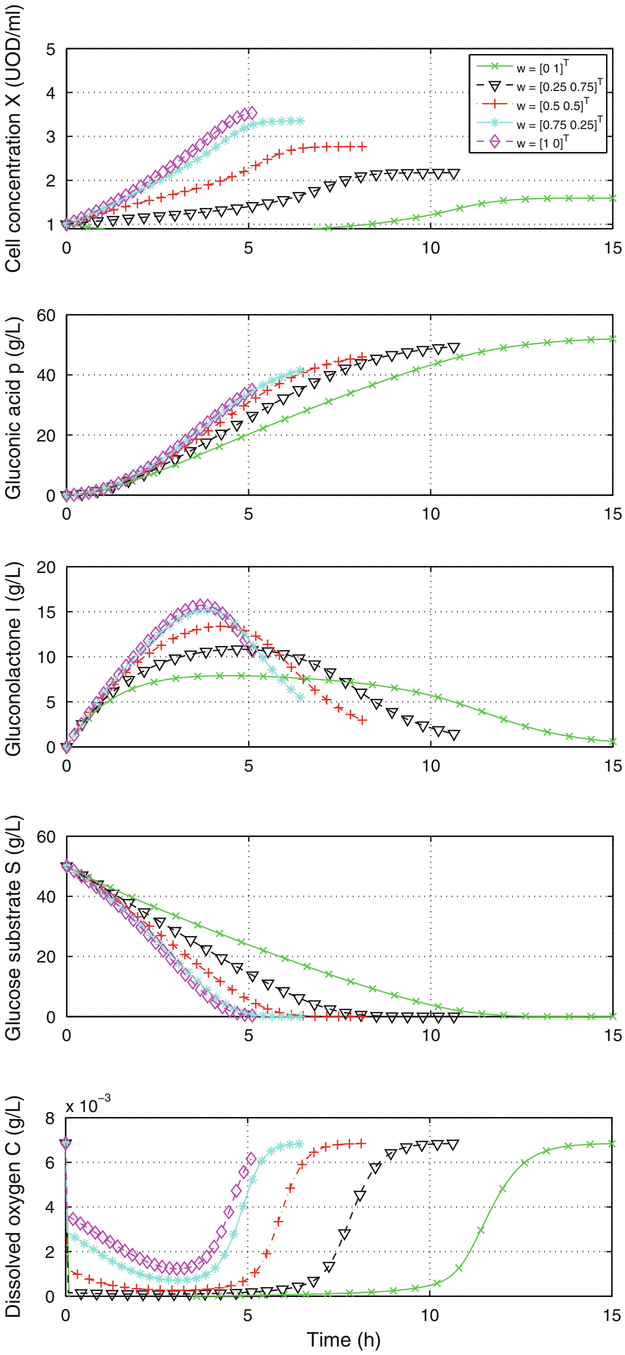

Fig. 3 Case 1: optimal states along the Pareto set obtained with NNC

Case 2

Figure 4 presents the Pareto frontier. As can be seen, the Pareto front exhibits very steep rises towards the individual minima. Hence, most of the trade-off is located in the so-called *knee* of the Pareto curve. Consequently, the Pareto points generated by the WS cluster in this region. On the other hand, NBI and (E)NNC are able to reproduce the Pareto set with a much more uniform spread.

Case 2 is the first case in which optimal time varying controls have to be found. For the current case, a control discretisation of 10 piecewise constant pieces is used for both controls. The optimal profiles for the substrate u_1 and inducer u_2 feed rate obtained with NBI are given in Fig. 5. As can be seen, in both optimal controls large singular arcs are present. Arcs are typically called *singular* or *sensitivity seeking* when they appear in an interval where no constraint on the control and/or states is active. Hence, the sensitivity of the objective with respect to the controls can be expected to be rather limited. Nevertheless, despite this limited sensitivity and a coarse control discretisation the same optimal cost values as mentioned in Ref. [33] can be obtained. However, the price to be paid is an increased number of SQP iterations. In general, the values for both controls remain quite low, especially when the minimisation of the inducer is concentrated on. When the

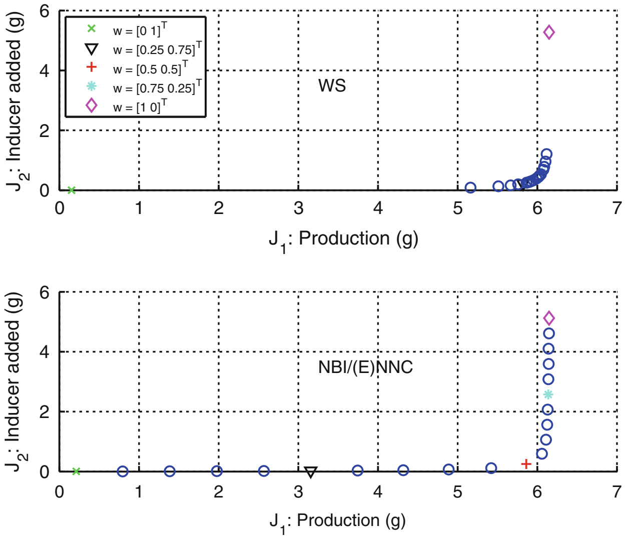

Fig. 4 Case 2: Pareto front with 21 points obtained with WS (*top*) and NBI/(E)NNC (*bottom*)

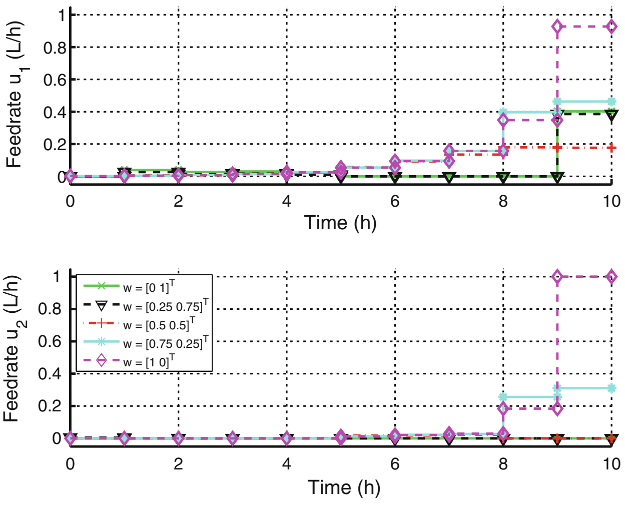

Fig. 5 Case 2: optimal controls along the Pareto set obtained with NBI

maximisation of the foreign protein production gets more and more priority, the inducer feed rates increase, in particular towards the end of the batch.

In view of brevity, only a selection of state profiles is depicted in Fig. 6, i.e., the amount of biomass $x_1(t) \cdot x_2(t)$ (g), the amount of foreign protein $x_1(t) \cdot x_4(t)$ (g) and the amount of inducer added J_2. Clearly, the biomass evolution does not differ much along the Pareto set. Typically an exponential evolution from the initially present 0.1 g to values between 26 and 31 g at the end is observed. Implications are that towards the end of the batch significantly more substrate has to be added to feed the micro-organisms and to counteract the dilution effect. Hence, in the beginning mainly biomass growth is important. However, when protein production is focussed on, inducer addition starts slowly around half of the batch time and increases significantly towards the end. This increase stimulates the available micro-organisms to produce the foreign protein and causes a boost in the total amount produced. The maximum amount of product is slightly higher than 6 g and requires about 5.1 g of inducer. Alternatively, inducer addition can be completely avoided (i.e., 0 g of inducer) but then the protein production is limited to only 0.2 g.

Case 3

Case 3 is the first one to tackle more than two objectives. In Fig. 7 the Pareto front is displayed for three objectives obtained with ENNC. The three individual minima as well as three intermediate Pareto optimal solutions are marked. Similar results can be obtained with NBI, but the results obtained with NNC slightly differ due to the different normalisation scheme (see Ref. [21] for more details). Results for the WS have been omitted as a highly non-uniform spread on the Pareto set was observed. To obtain a better understanding of the trade-off between the different objectives, three pairwise Pareto fronts depict each of the three possible pairwise objective combinations. Based on these plots, it is seen that trade-offs between the substrate added (i.e., J_3) on the one hand and productivity (i.e., J_1) and production (i.e., J_2) on the other hand, are rather linear. However, the trade-off between these last two appears to be more curved. Case 3 was posed as to maximise production in Ref. [4, 22] without looking at other objectives. However, identical values as in Ref. [22] for optimal production have been obtained (i.e., 20,841.2 g), which outperform the values mentioned in Ref. [4].

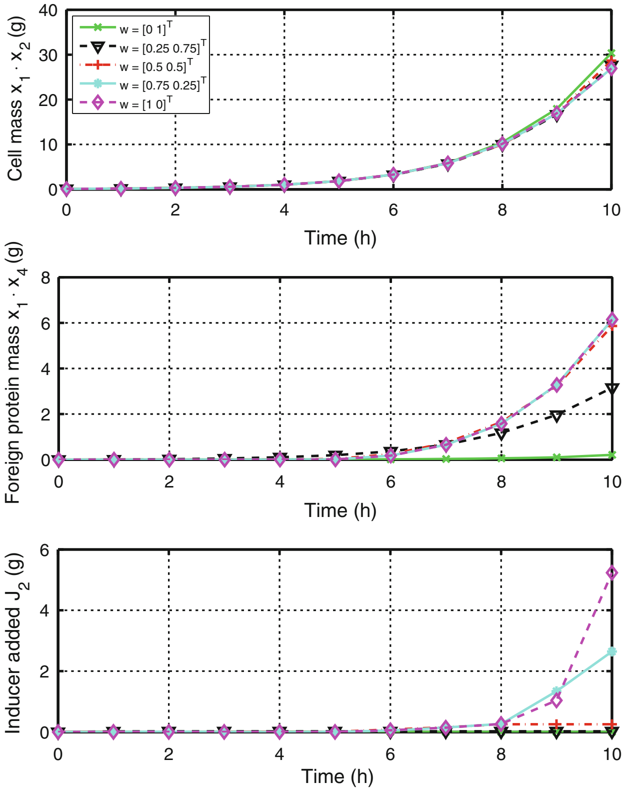

Fig. 6 Case 2: optimal states along the Pareto set obtained with NBI

The optimal controls and a selection of the optimal states corresponding to the marked points are displayed in Figs. 8, 9. To maximise productivity, again short batches are preferred with high feeding rates in the beginning followed by a short batch phase at the end. This strategy typically boosts the biomass growth and the production rate but results in rather low production amounts and does not care about the amount of substrate required. To maximise the production the available amount of substrate is added more carefully and the batch time

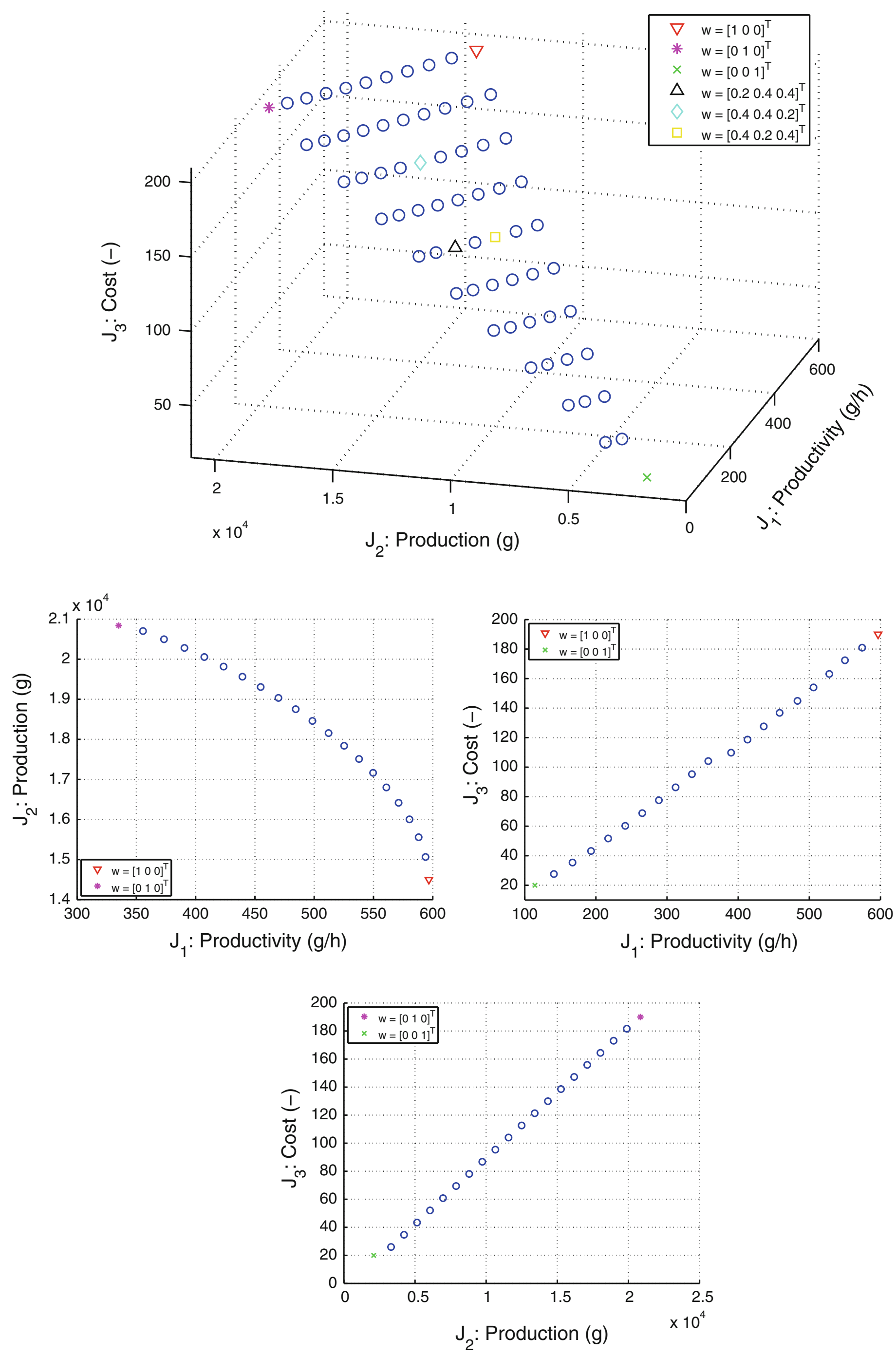

Fig. 7 Case 3: Pareto fronts for the three objectives together with 66 points and the three pairwise objectives with 21 points obtained with ENNC

increases. In particular, a large singular feeding phase which increases towards the end is present. The increase is due to the biomass increase and the dilution effect. When concentrating on the economic use of substrate, only the minimum amount of 30 L is fed in an intermediate batch time. This yields a rather small amount of product and a low productivity. It is seen that the intermediate Pareto optimal points exhibit an intermediate behaviour for the control and states.

Case 4

The Pareto front for maximising both the production and the purity is shown in Fig. 10. The WS does not succeed in presenting a nice approximation of the Pareto set as points tend to cluster around the maximum production point. Hence, these results have been omitted. It can also be seen that the trade-offs are not large. The production ranges between 267 and 287 g, while the purity varies only

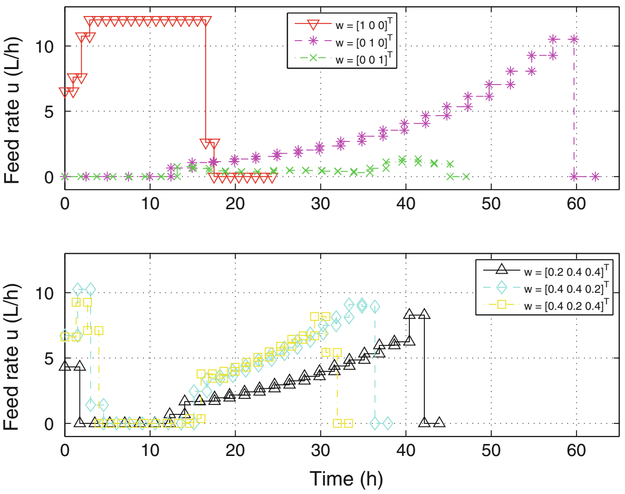

Fig. 8 Case 3: optimal controls along the Pareto set obtained with ENNC

between 1.44 and 1.48 g/L. Whether or not these differences are significant in practice has to be decided by the decision maker. When the minimum production constraint is removed, a comparison with the maximum purity results from Ref. [35] can be made. In that case, the reported value of 1.68 g/L is found. However, the production is then 260.71 g.

The resulting controls are displayed in Fig. 11. They exhibit most often a *singular-maximum-minimum* structure. When focussing on purity, the singular arc and the maximum arc are the shortest, while the minimum arc is the largest. When shifting towards production, mainly the minimum arc decreases, while the singular and the maximum arc gradually increase. When the entire emphasis is put on production, the last minimum part vanishes and a constrained control appears, which keeps the biomass constant at its upper limit.

A selection of the states is displayed in Fig. 12. When production is focussed on, the biomass constraint is active from 90 h and remains active until the end of the batch, while for the other cases this constraint is only active at the batch end. The differences along the batch duration are small for both the amount of product made and the product concentration. A zoom near the batch end elucidates these small differences.

Discussion and computational expense

Table 4 gives an overview of (1) the features of the different multiple objective optimal control problems, (2) the algorithmic settings used as well as (3) the computational expense (number of SQP iterations, CPU times and average CPU time per Pareto point). All computations have been performed on a PC with a 1.86 GHz processor and 2 GB

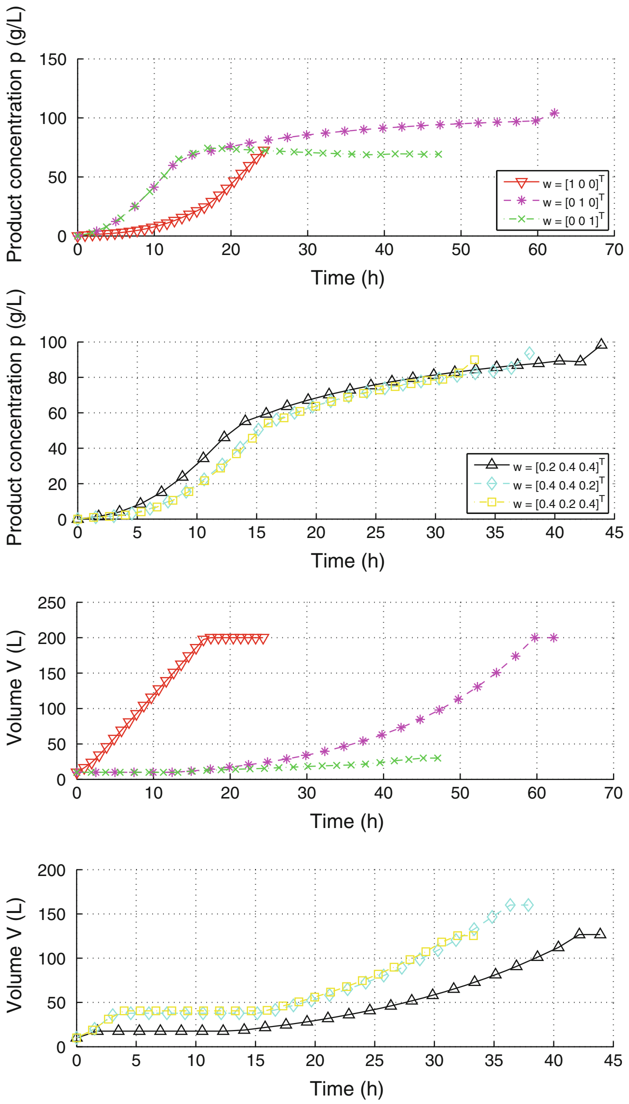

Fig. 9 Case 3: optimal states along the Pareto set obtained with ENNC

RAM memory. Tight integration tolerances have been selected to ensure an accurate computation of the profiles and their sensitivities with respect to the degrees of freedom to be determined. These sensitivities can be low on, e.g., a singular arc. Optimality tolerances were chosen such that tightening the tolerances did not improve the Pareto sets or the optimal control profiles. NBI and (E)NNC induce a similar computational burden, which is higher than the one for WS, due to the additional equality and inequality constraints. However, the Pareto sets generated by WS in general do not achieve the same accuracy. Pareto points are missed due to bad objective scaling and low sensitivity. The average computation times per Pareto point vary between 0.2 and 7 s. Cases 2, 3 and 4 exhibit singular arcs in the solutions, which maybe difficult to optimise accurately due to the low sensitivity. The hot-starting

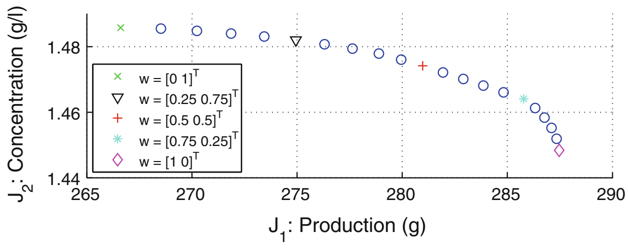

Fig. 10 Case 4: Pareto front with 21 points obtained with NBI

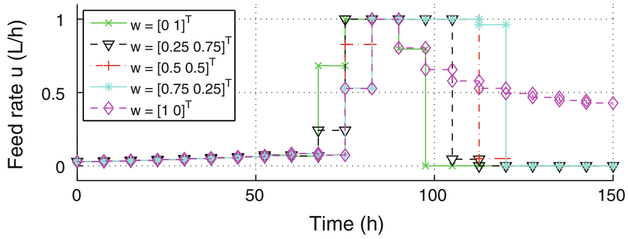

Fig. 11 Case 4: optimal controls along the Pareto set obtained with NBI

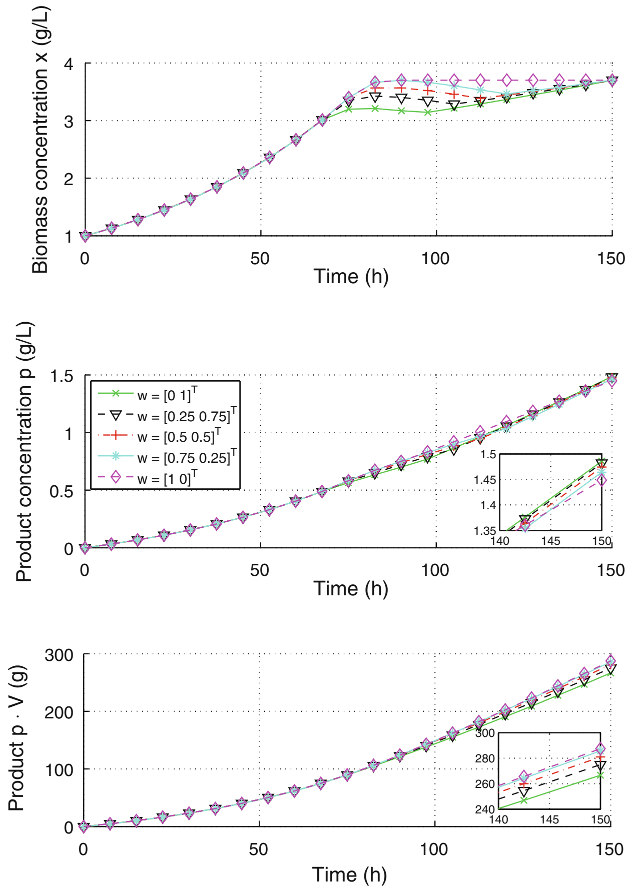

Fig. 12 Case 4: optimal states along the Pareto set obtained with NBI

strategy has been found to speed up computations by a factor around 2. Possible extensions involve the incorporation of methods for of integer controls [20].

In summary, the multiple objective optimal control problems considered exhibit features as non-linear dynamics, fixed/free initial conditions, path and terminal constraints, singular/constrained control arcs, fixed/free end times. Hence, it has been shown that the ACADO Multi-Objective Toolkit is able to efficiently produce accurate Pareto sets for general multi-objective dynamic optimisation problems in bioprocess engineering.

As requested by one of the reviewers, an illustration of an algorithm from a different class is provided. The cases 1 and 4 have also been solved with the multi-objective evolutionary algorithm NSGA-II [8]. The C source code has been obtained from the Kanpur Genetic Algorithms Laboratory website [15]. The selected implementation allows for both real and binary variables implementation and is able to include constraints. This algorithm has been coupled to the ACADO integrators, which are exploited to simulate the process behaviour each time. Recommended and default settings have been adopted (e.g., the muation frequency equals 1 over the number of decision variables). The algorithmic settings and the computational burden are summarised in Table 5. The resulting Pareto sets are depicted together with the ACADO Multi-objective results in Fig. 13. Computation times mentioned involve only the time for integrating the model equations, as the time for the NSGA-II algorithm itself is considered to be significantly lower than the time for the integrations. For case 1 an almost identical Pareto set has been obtained with a nice spread. Since only four decision variables have to be found and only simple bounds are specified for these decision variables, the optimisation runs efficiently resulting in a lower computation time than ACADO Multi-objective. It has to be mentioned that although ACADO Multi-objective only uses local optimisation routines, no problems with local minima have been experienced. ACADO Multi-objective is able to get to the same extreme points as obtained with the genetic algorithm, which is generally regarded as a global optimisation routine. For case 4, the results are different. Clearly 10,000 instead of 1,000 generations are needed to converge to the Pareto front. As the control is discretised with 20 piecewise constant of equal length, 20 degrees of freedom have to optimised. Also a state constraint is present. In this case NSGA-II requires significantly more computation time per Pareto point. Also the areas near the individual minima are not well covered. The singular arcs are as accurately determined as before, due to the low sensitivity of the cost with respect to changes in these arcs.

In summary, the NSGA-II is easily and flexibly coupled to an existing process simulator, while in ACADO Multi-

Table 5 Overview of algorithmic settings for NSGA-II and computational expense

	Case 1	Case 4	
# Pareto points	40	400	400
# Generations	100	1000	10000
$p_{cross\text{-}over}$	0.75	0.70	0.70
$p_{mutation}$	0.25	0.05	0.05
$\eta_{cross\text{-}over}$	15	15	15
$\eta_{mutation}$	25	25	25
Total CPU [s] (entire front)	30.0	1421.6	15637.6
Average CPU [s] (1 point)	0.75	3.55	39.09

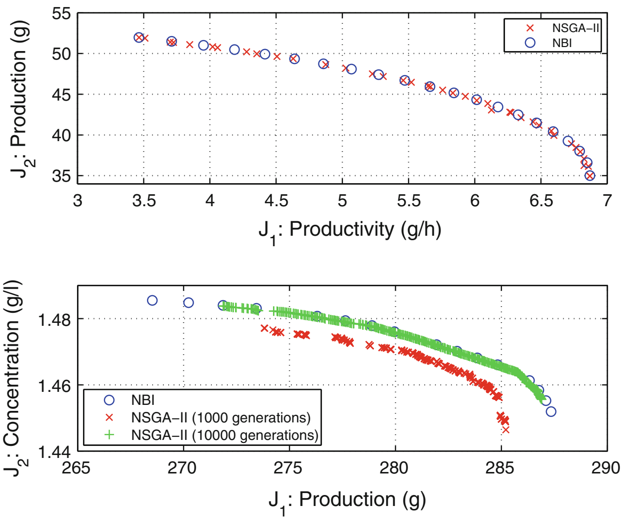

Fig. 13 Pareto set for NBI and NSGA-II: case 1 (top) and case 4 (bottom)

objective a model has to be re-coded. NSGA-II has more difficulties to deal with (1) larger numbers of decision variables (e.g., due to fine uniform control discretisations) and (2) other constraints than simple bounds on the decision variables (e.g., due to state constraints). Finally, it has to be emphasised that these experiments with the NSGA-II algorithm have been performed by the authors from a non-experienced user perspective. Experienced NSGA-II users will be able to tune the algorithm allowing for performance improvements.

Conclusion

This paper deals with the fast and efficient solution of biochemical optimal control problems with multiple objectives. To this end, several scalarisation techniques for multi-objective optimisation, e.g., WS, NBI and (E)NNC have been integrated with fast deterministic direct optimal control approaches (e.g., multiple shooting). All techniques have been implemented in the ACADO Multi-Objective Toolkit, which is available at http://www.acadotoolkit.org. The toolkit has been succesfully evaluated on four bioreactor cases from literature. The objective values in the resulting Pareto sets are added as supplementary material to this paper.

Acknowledgments Work supported in part by KULeuven: OT/10/035, OPTEC (Center-of-Excellence Optimization in Engineering PFV/10/002), SCORES4CHEM (KP/09/005), GOA/10/09 MaNet; by FWO: G.0320.08 (convex MPC), G. 0558.08 (Robust MHE), G.0377.09 (Mechatronics MPC); by IWT: SBO LeCoPro; by the Belgian Federal Science Policy Office: Belgian Program on Interuniversity Poles of Attraction; by EU: FP7-HD-MPC (INFSO-ICT-223854), FP7-EMBOCON (ICT-248940), FP7-SADCO (MC ITN-264735), ERC HIGHWIND (259 166) and by contract research ACCM. D. Telen has a Ph.D. grant of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen). J.F. Van Impe holds the chair Safety Engineering sponsored by the Belgian chemistry and life sciences federation essenscia.

References

1. Balsa-Canto E, Banga J, Alonso A, Vassiliadis V (2001) Dynamic optimization of chemical and biochemical processes using restricted second-order information. Comput Chem Eng 25(4–6):539–546
2. Biegler L (1984) Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. Comput Chem Eng 8:243–248
3. Bock H (1983) Recent advances in parameter identification techniques for ODE. In: Deuflhard P, Hairer E (eds). Numerical treatment of inverse problems in differential and integral equations. Birkhäuser, Boston, pp 95–121
4. Chen C, Hwang C (1990) Optimal control computation for differential algebraic process systems with general constraints. Chem Eng Commun 97(1):9–26
5. Das I, Dennis J (1997) A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. Struct Optim 14:63–69
6. Das I, Dennis J (1998) Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems. SIAM J Optim 8:631–657
7. Deb K (2001) Multi-objective optimization using evolutionary algorithms. Wiley, London
8. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6:181–197
9. Eichfelder G (2008) Adaptive scalarization methods in multiobjective optimization. Vector optimization. Springer, Berlin
10. Griewank A (1989) On automatic differentiation. In: Iri M, Tanabe K (eds) Mathematical programming: recent developments and applications. Kluwer Academic, Amserdam, pp 83–108
11. Griewank A, Walther A (2008) Evaluating derivatives: principles and techniques of algorithmic differentiation. SIAM, Philadelphia
12. Halsall-Whitney H, Taylor D, Thibault J (2003) Multicriteria optimization of gluconic acid production using net flow. Bioprocess Biosyst Eng 25:299–307
13. Hirmajer T, Balsa-Canto E, Banga JR (2009) DOTcvpSB, a software toolbox for dynamic optimization in systems biology. BMC Bioinf 10(1):199–212

14. Houska B, Ferreau H, Diehl M (2011) ACADO Toolkit—an open-source framework for automatic control and dynamic optimization. Optim Control Appl Methods 32:298–312
15. Kanpur Genetic Algorithm Laboratory: http://www.iitk.ac.in/kangal/codes.shtml
16. Lang YD, Biegler L (2007) A software environment for simultaneous dynamic optimization. Computers and Chemical Engineering 31:931–942
17. Leineweber D, Bauer I, Bock H, Schlöder J (2003) An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization. Part I. Comput Chem Eng 27:157–166
18. Leineweber D, Schäfer A, Bock H, Schlöder J (2003) An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization. Part II. Comput Chem Eng 27:167–174
19. Logist F, Houska B, Diehl M, Van Impe J (2010) Fast pareto set generation for nonlinear optimal control problems with multiple objectives. Struct Multidisciplinary Optim 42:591–603
20. Logist F, Sager S, Kirches C, Van Impe J (2010) Efficient multiple objective optimal control of dynamic systems with integer controls. J Process Control 20:810–822
21. Logist F, Van Impe J (2012) Novel insights for multi-objective optimisation in engineering using normal boundary intersection and (enhanced) normalised normal constraint. Struct Multidisciplinary Optim 45:417–431
22. Luus R (1993) Application of dynamic programming to differential-algebraic process systems. Comput Chem Eng 17(4):373–377
23. Maeda K, Fukano Y, Yamamichi S, Nitta D, Kurata H (2011) An integrative and practical evolutionary optimization for a complex, dynamic model of biological networks. Bioprocess Biosyst Eng 34:433–446
24. Mandal C, Gudi R, Suraishkumar G (2005) Multi-objective optimization in aspergillus niger fermentation for selective product enhancement. Bioprocess Biosyst Eng 28:149–164
27. Messac A, Ismail-Yahaya A, Mattson C (2003) The normalized normal constraint method for generating the Pareto frontier. Struct Multidisciplinary Optim 25:86–98
28. Messac A, Mattson C (2004) Normal constraint method with guarantee of even representation of complete Pareto frontier. AIAA J 42:2101–2111
29. Miettinen K (1999) Nonlinear multiobjective optimization. Kluwer Academic, Boston
25. Process System Enterprise Limited: gPROMS (2010)
30. Sanchis J, Martinez M, Blasco X, Salcedo J (2008) A new perspective on multiobjective optimization by enhanced normalized normal constraint method. Struct Multidisciplinary Optim 36:537–546
31. Sargent R, Sullivan G (1978) The development of an efficient optimal control package. In: Stoer J (eds). Proceedings of the 8th IFIP Conference on Optimization Techniques. Springer, Heidelberg, pp 158–168
32. Sarkar D, Modak J (2004) Genetic algorithms with filters for optimal control problems in fed-batch bioreactors. Bioprocess Biosyst Eng 26:295–306
33. Sarkar D, Modak J (2005) Pareto-optimal solutions for multi-objective optimization of fed-batch bioreactors using nondominated sorting genetic algorithm. Chem Eng Sci 60:481–492
34. Schlegel M, Stockmann K, Binder T, Marquardt W (2005) Dynamic optimization using adaptive control vector parameterization. Comput Chem Eng 29:1731–1751
35. Srinivasan B, Palanki S, Bonvin D (2003) Dynamic optimization of batch processes I. Characterization of the nominal solution. Comput Chem Eng 27:1–26
36. Tebbani S, Dumur D, Hafidi G (2008) Open-loop optimization and trajectory tracking of a fed-batch bioreactor. Chem Eng Process 47:1933–1941
37. Thibault J, Taylor D, Fonteix C (2001) Multicriteria optimization for the production of gluconic acid. In: 8th International conference on computer applications in biotechnology, pp 287–292
38. Tholudur A, Ramirez W (1997) Obtaining smoother singular arc policies using a modified iterative dynamic programming algorithm. Int J Control 68:1115–1128
26. Tomlab Optimization Inc (2010) PROPT—Matlab optimal control software
39. Zhou Y, Titchener-Hooker N (2003) The application of a Pareto optimisation method in the design of an integrated bioprocess. Bioprocess Biosyst Eng 25:349–355