Journal of Biotechnology 383 (2024) 94–102

Contents lists available at ScienceDirect

Journal of Biotechnology

journal homepage: www.elsevier.com/locate/jbiotec

OptMSP: A toolbox for designing optimal multi-stage (bio)processes

Jasmin Bauer, Steffen Klamt *

Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, Magdeburg, Germany

ARTICLE INFO

Keywords:
Process optimization
Bioprocess engineering
Multi-stage bioprocess
Escherichia coli
Python toolbox
Volumetric productivity

ABSTRACT

One central goal of bioprocess engineering is to maximize the production of specific chemicals using microbial cell factories. Many bioprocesses are one-stage (batch) processes (OSPs), in which growth and product synthesis are coupled. However, OSPs often exhibit low volumetric productivities due to the competition for substrate for biomass and product synthesis implying trade-offs between biomass and product yields. Two-stage or, more generally, multi-stage processes (MSPs) offer the potential to tackle this trade-off for improved efficiency of bioprocesses, for example, by separating growth and production. MSPs have recently gained much attention, also because of a rapidly growing toolbox for the dynamic control of metabolic fluxes. Despite these promising advancements, computational tools specifically tailored for the optimal design of MSPs in the field of biotechnology are still lacking.

Here, we present OptMSP, a new Python-based toolbox for identifying optimal MSPs maximizing a user-defined process metrics (such as volumetric productivity, yield, and titer or combinations thereof) under given constraints. In contrast to other methods, our framework starts with a set of well-defined modules representing relevant stages or sub-processes. Experimentally determined parameters (such as growth rates, substrate uptake and product formation rates) are used to build suitable ODE models describing the dynamic behavior of each module. OptMSP finds then the optimal combination of those modules, which, together with the optimal switching time points, maximize a given objective function. We demonstrate the applicability and relevance of the approach with three different case studies, including the example of lactate production by *E. coli* in a batch setup, where an aerobic growth phase can be combined with anaerobic production phases with or without growth and with or without enhanced ATP turnover.

1. Introduction

In microbial fermentation processes, microorganisms are used and designed to produce specific chemicals and compounds with high titer, productivity (rate) and yield (TRY metrics). Many batch bioprocesses follow a one-stage process (OSP) approach, coupling growth and product synthesis. However, this coupling often leads to low volumetric productivities due to shared substrate usage for biomass and product synthesis, resulting in trade-offs between biomass and product yields. To address this problem, two-stage or multi-stage processes[1] (MSPs) are being explored as a means to enhance bioprocess productivity, e.g. by separating growth and production phases (see Fig. 1). Apart from switching from a growth to a production phase, other reasons may favor the use of MSPs as bioprocess strategies, for example, switching from batch to fed-batch operation to avoid high initial substrate concentrations that could induce inhibitory effects or overflow metabolism.

Theory and application of MSPs have garnered significant attention in recent years (Burg et al., 2016; Raj et al., 2020; Gadkar et al., 2005; Anesiadis et al., 2013; Hjersted and Henson, 2006; Jabarivelisdeh and Waldherr, 2018; Ryu et al., 2019; Klamt et al., 2018), also because of the growing toolbox for dynamic control of metabolic fluxes via genetic switches that depend, for example, on cell density (quorum sensing) (Gupta et al., 2017; Soma and Hanai, 2015), temperature (Harder et al., 2018), phosphate (Menacho-Melgar et al., 2020; Farmer and Liao, 2000), oxygen (Hwang et al., 2017; Wichmann et al., 2023), or light (Carrasco-López et al., 2020). Despite these advancements, there is still a need for computational tools to efficiently design optimal MSPs (Burg et al., 2016).

* Corresponding author.
E-mail address: klamt@mpi-magdeburg.mpg.de (S. Klamt).

[1] In the literature, it is sometimes distinguished between two-phase or multi-phase processes (different phases in one bioreactor setup) and two-stage or multi-stage processes (different phases in distinct bioreactors or vessels). Herein, we do not make such a distinction, i.e. the terms two-stage or multi-stage process include both variants.

https://doi.org/10.1016/j.jbiotec.2024.01.009
Received 9 November 2023; Received in revised form 17 January 2024; Accepted 23 January 2024
Available online 5 February 2024

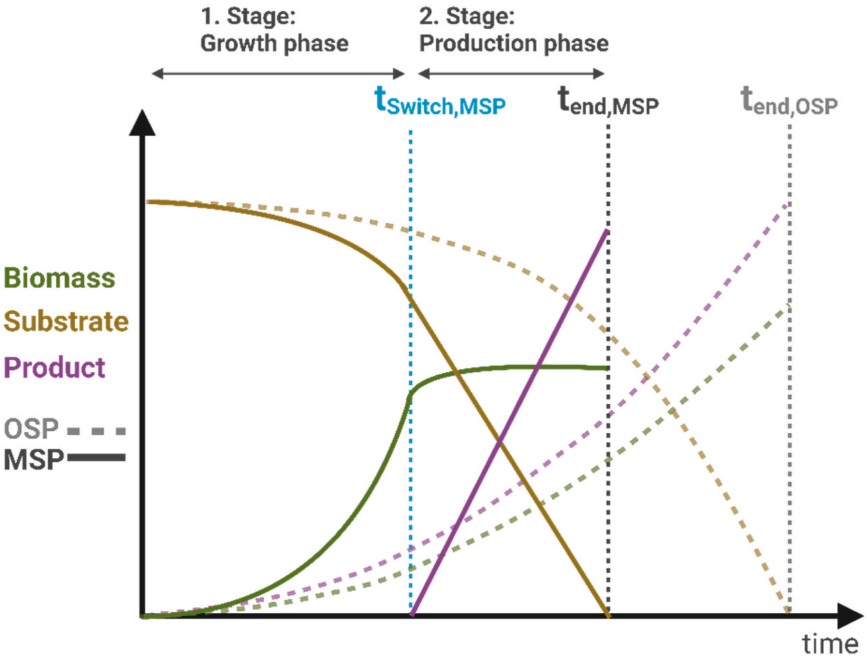

Fig. 1. MSP vs OSP. Comparison of a typical MSP, here represented as a two-stage process with separate growth and production phase, and an OSP. The MSP generates higher volumetric productivity through the focus on biomass formation in the first stage enabling faster product formation in the second phase. The OSP produces product and biomass simultaneously, which results in a shared usage of substrate and therefore slower growth compared to the first phase of the MSP. While the final product titer of the OSP might be higher compared to the MSP, its volumetric productivity is lower because of the longer duration of the process. (Figure created with BioRender.com.).

Various factors demand consideration when enhancing bioprocesses within multi-stage setups. First of all, what is to be optimized (e.g. volumetric productivity or product yield) under which constraints. Second, what are the phases/stages that can be used and how to design them to make the overall process efficient. For example, are there strategies that can effectively counterbalance the adverse impact of a declining substrate uptake rate often observed during growth-arrested production phases, a phenomenon that can even lead to an underperformance of MSPs compared to OSPs (Klamt et al., 2018). Finally, how and when (optimal switching time point) will the transition from one to another stage be initiated, in particular, which specific process conditions (e.g. oxygen supply or limitation of nutrients such as nitrogen or phosphate) will be changed or/and which genetic switches and targets for metabolic engineering will be used for achieving the intended decoupling of growth and production phases. The multiplicity and complexity of those factors emphasize the need for a computational approach to optimize MSPs.

A smaller number of tools related to the design and optimization of MSP have been presented in the literature. For example, mcPECASO (Raj et al., 2020) can be used to find optimal two-stage processes based on approximations of the metabolic behavior of the production host. MoVE (Venayak et al., 2018) is a computational approach for identifying metabolic valves that can be targeted for switching between growth and production, however, it does not support the design of the actual process (e.g. determination of the optimal time point for switching between different phases).

Here, we introduce the toolbox OptMSP (Optimization of Multi-Stage Processes), which can be used to identify, from a given set of possible stages, optimal MSPs in a batch setup maximizing a user-defined process metrics under given constraints (e.g., demanded minimum product yield). In contrast to other methods, our framework starts with a set of well-defined modules representing relevant stages or sub-processes (e.g., aerobic growth, anaerobic production with or without growth). The dynamic behavior of each module needs to be described via mathematical models (e.g. based on ordinary differential equations (ODE)), using parameters such as growth rate or substrate uptake and product formation rates determined in separate experiments. OptMSP finds then the optimal combination of those modules, which, together with the optimal switching time points, maximize a given objective function. OptMSP is a Python-based (VanRossum and Drake, 2010) toolbox and provides various options to design optimal MSP based on the given process modules. We demonstrate the applicability and relevance of the approach with three different case studies, including the example of lactate production by *E. coli* in a batch setup, where an aerobic growth phase can be combined with anaerobic production phases, the latter with or without growth arrest and with or without enforced ATP wasting.

2. Methods

2.1. Overview: Workflow of OptMSP

The workflow of OptMSP is depicted in Fig. 2 and is explained in detail in the following.

2.1.1. Step 1: specification of process modules

Initially, all sub-processes (modules) considered to be possible phases/stages of the MSP are defined together with their specific conditions (e.g. oxygen availability, pH, temperature, gene inducer(s)). The process behavior of each module (including the metabolic phenotype of the microbial production host) is ideally determined in separate experiments, where, for example, time courses of substrates, products and

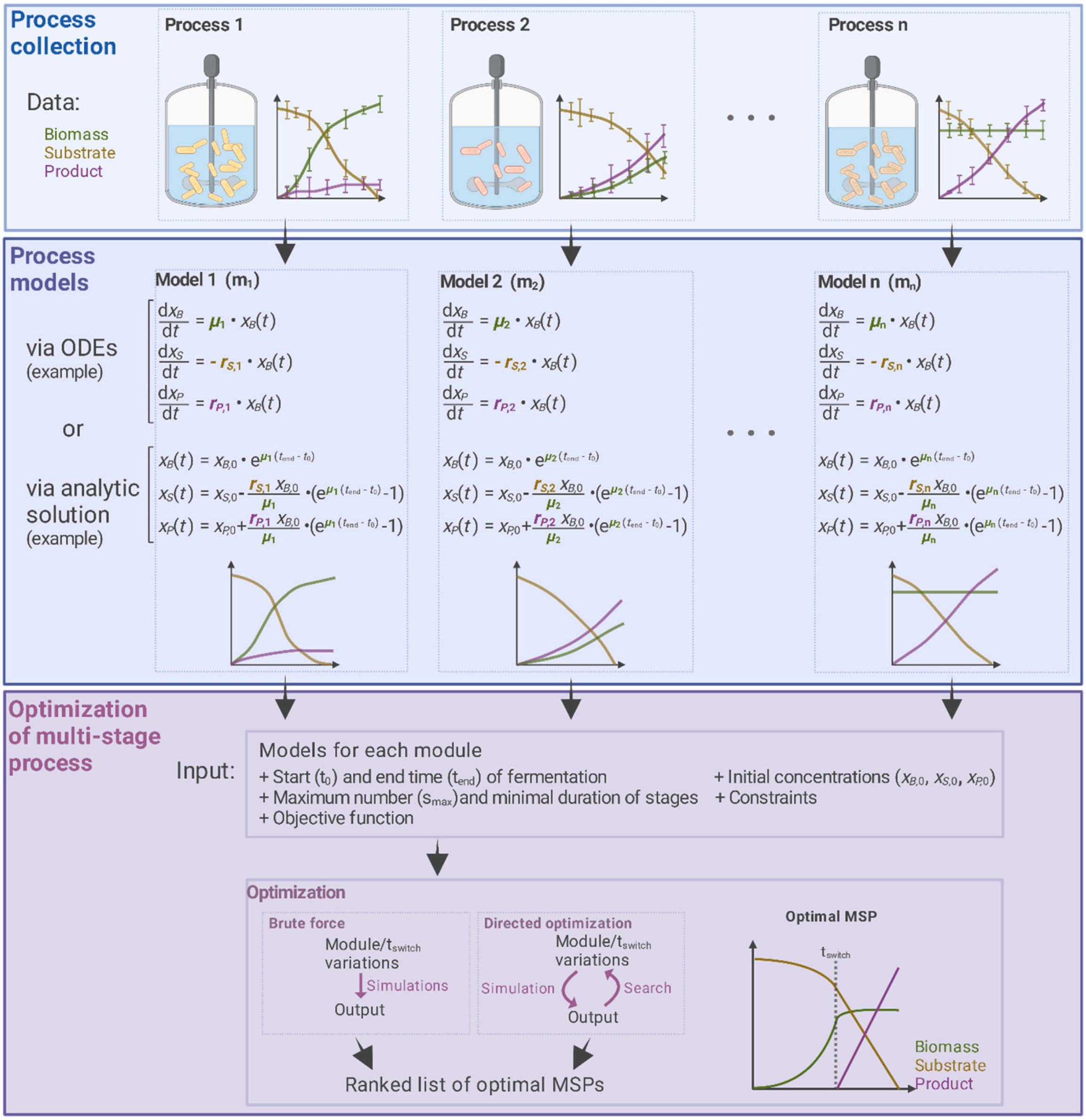
Process collection
Data:
Biomass
Substrate
Product
Process 1
Process 2
Process n
Process models
via ODEs (example)
or
via analytic solution (example)
Model 1 (m1)
Model 2 (m2)
Model n (mn)
$\frac{dx_B}{dt} = \mu_1 \cdot x_B(t)$
$\frac{dx_S}{dt} = -r_{S,1} \cdot x_B(t)$
$\frac{dx_P}{dt} = r_{P,1} \cdot x_B(t)$
$x_B(t) = x_{B,0} \cdot e^{\mu_1 (t_{end} - t_0)}$
$x_S(t) = x_{S,0} - \frac{r_{S,1} x_{B,0}}{\mu_1} \cdot (e^{\mu_1 (t_{end} - t_0)} - 1)$
$x_P(t) = x_{P,0} + \frac{r_{P,1} x_{B,0}}{\mu_1} \cdot (e^{\mu_1 (t_{end} - t_0)} - 1)$
Optimization of multi-stage process
Input:
Models for each module
+ Start (t0) and end time (tend) of fermentation
+ Maximum number (smax) and minimal duration of stages
+ Objective function
+ Initial concentrations ($x_{B,0}$, $x_{S,0}$, $x_{P,0}$)
+ Constraints
Optimization
Brute force
Module/tswitch variations
Simulations
Output
Directed optimization
Module/tswitch variations
Simulation
Search
Output
Ranked list of optimal MSPs
Optimal MSP
tswitch
Biomass
Substrate
Product

Fig. 2. Workflow of OptMSP. For details see text. (Figure created with https://biorender.com.).

biomass under the respective conditions are used to calculate parameters such as the specific rates of growth (μ), substrate uptake (r_S) and product formation (r_P).

2.1.2. Step 2: Process models

For each of the n modules, suitable process models are constructed making use of measurements and parameters determined in the experiments in step 1. The models are formulated as ODEs:

$$\frac{d\mathbf{x}(t)}{dt} = f_m(\mathbf{x}(t)) \quad m = 1, .., n. \tag{1}$$

$\mathbf{x}$ is the vector of the state variables and f_m describes the ODEs of the m-th module. In theory, the user can incorporate an arbitrary number of state variables in the models, but at least biomass, substrate, and product should be included to allow for the calculation of TRY performance measures. A specific MSP is then defined by (i) an ordered set $z = \{z_1, z_2, \ldots, z_j\}$ of j selected modules, (ii) a set $t_{switch} = \{t_{switch,1}, t_{switch,2}, \ldots, t_{switch,j-1}\}$ containing the $j-1$ switching times linking the j modules of z ($t_{switch,1} < t_{switch,2} < \ldots < t_{switch,j-1}$), (iii) the start ($t_0$) and (maximal) end time (t_{end}) of the process, and (iv) the initial state variables $\mathbf{x_0}$. The entire MSP can then be simulated as a sequential simulation of its modules

$$\begin{aligned}
&\frac{d\mathbf{x}(t)}{dt} = f_{z_1}(\mathbf{x}(t)),\ \mathbf{x}(t_0) = \mathbf{x_0},\ t \in [t_0, t_{switch,1}] \\
&\frac{d\mathbf{x}(t)}{dt} = f_{z_2}(\mathbf{x}(t)),\ \mathbf{x}(t_{switch,1}) = \mathbf{x_1},\ t \in [t_{switch,1}, t_{switch,2}] \\
&\qquad\vdots \\
&\frac{d\mathbf{x}(t)}{dt} = f_{z_j}(\mathbf{x}(t)),\ \mathbf{x}(t_{switch,j-1}) = \mathbf{x_{j-1}},\ t \in [t_{switch,j-1}, t_{end}]
\end{aligned} \tag{2}$$

where $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x_{j-1}}$ represent the initial values of the second, third, …, j-th simulated module, which are taken from the simulation of the previous module ($\mathbf{x_1} = \mathbf{x}(t_{switch,1})$, $\mathbf{x_2} = \mathbf{x}(t_{switch,2})$, etc.). Importantly, it may happen that the actual process finishes before t_{end} is reached (e.g., if the substrate is depleted). We here assume that the simulation of the MSP performed as described above also delivers the actual time point where the process is finished, denoted by $\widehat{t}_{end}$ (note that $\widehat{t}_{end} = t_{end}$ if the process is not finished before (or at) t_{end}). For example, if the process is considered to be finished when the substrate is depleted then $\widehat{t}_{end}$ is the time point where the substrate concentration reaches a concentration of zero (or where it falls below a given threshold concentration).

While our approach allows the use of arbitrary (complex) ODE models, herein we will also consider the case that the ODE model of a module m is a simple process model for exponential growth solely based on the concentrations of biomass (x_B), substrate (x_S) and product (x_P) together with the associated specific growth (μ_m), substrate uptake ($r_{S,m}$) and product formation ($r_{P,m}$) rate of the production host (see also Fig. 2):

$$\frac{\mathrm{d}x_B(t)}{\mathrm{d}t} = \begin{cases} \mu_m x_B(t) & \text{if } x_s(t) > 0 \\ 0 & \text{if } x_s(t) = 0 \end{cases} \tag{3}$$

$$\frac{\mathrm{d}x_S(t)}{\mathrm{d}t} = \begin{cases} -r_{S,m} x_B(t) & \text{if } x_S(t) > 0 \\ 0 & \text{if } x_S(t) = 0 \end{cases} \tag{4}$$

$$\frac{\mathrm{d}x_P(t)}{dt} = \begin{cases} r_{P,m} x_B(t) & \text{if } x_S(t) > 0 \\ 0 & \text{if } x_S(t) = 0 \end{cases} \tag{5}$$

Such a simplistic model reflects only the exponential phase of a batch process while it would neglect the reduced metabolic activity in the lag and (beginning) stationary phase. However, lag and stationary phase may only be relevant for smaller time periods of the first and the last stage of an MSP, respectively. Using a simplified model with only exponential phase representation may serve as a first approximation and can even be solved analytically as described in the following. The time point $\widehat{t}_{\text{end}}$ where the substrate would be depleted in the given module can be calculated as $\widehat{t}_{\text{end}} = t_0 + \ln\left(\frac{\mu_m x_{S,0}}{r_{S,m} x_{B,0}} + 1\right)/\mu_m$. With this, the analytical solutions of the dynamic process behavior of module m read:

$$x_B(t) = \begin{cases} x_{B,0} \cdot e^{\mu_m (t - t_0)} & \text{if } t < \widehat{t}_{\text{end}} \\ x_{B,0} \cdot e^{\mu_m (\widehat{t}_{\text{end}} - t_0)} & \text{if } t \geq \widehat{t}_{\text{end}} \end{cases} \tag{6}$$

$$x_S(t) = \begin{cases} x_{S,0} - \dfrac{r_{S,m} x_{B,0}}{\mu_m} \cdot \left(e^{\mu_m (t - t_0)} - 1\right) & \text{if } t < \widehat{t}_{\text{end}} \\ 0 & \text{if } t \geq \widehat{t}_{\text{end}} \end{cases} \tag{7}$$

$$x_P(t) = \begin{cases} x_{P,0} + \dfrac{r_{P,m} x_{B,0}}{\mu_m} \cdot \left(e^{\mu_m (t - t_0)} - 1\right) & \text{if } t < \widehat{t}_{\text{end}} \\ x_{P,0} + \dfrac{r_{P,m} x_{B,0}}{\mu_m} \cdot \left(e^{\mu_m (\widehat{t}_{\text{end}} - t_0)} - 1\right) & \text{if } t \geq \widehat{t}_{\text{end}} \end{cases} \tag{8}$$

$x_{B,0}$, $x_{S,0}$ and $x_{P,0}$ represent the start concentrations of biomass, substrate and product, respectively, and t_0 the start time of the process. The analytical solution of a combination of different (analytically solved) modules can then be obtained in an analogous scheme as for the general ODEs described above in Eq. (2).

2.1.3. Step 3: finding the optimal MSP

The last step of the workflow is the optimization itself, in which the OptMSP package searches for the optimal MSP based on (i) the modules with their dynamic models constructed in step 2, (ii) a user-defined objective function h, (iii) additional constraints (e.g. demanded minimum product yields or titers) and (iv) other metaparameters of the optimization routine (e.g., the maximal number of combinable stages in the MSP; see below). The general form of the MSP optimization problem considered herein reads

$$\max_{z, t_{switch}} h\left(\mathbf{x}(t_0),\ \mathbf{x}(\widehat{t}_{\text{end}})\right)$$

$$s.t.$$

Eq. (2)

$$z \in V_n^k,\quad 1 \leq k \leq s_{\max}$$

$$t_0 < t_{switch,1} < \ldots < t_{switch,|z|-1} < t_{end}$$

$$\mathbf{g}\left(\mathbf{x}(t_0), \mathbf{x}(\widehat{t}_{end})\right) \leq \mathbf{b} \tag{9}$$

Here, z is an ordered set of stages representing any possible k-variation of all n modules (denoted by V_n^k) with k varying between 1 and $s_{\max}$, where $s_{\max}$ denotes the maximum number of combinable stages. $\boldsymbol{t_{switch}}$ is a suitable vector with $|z| - 1$ switching times connecting the chosen stages of the MSP. The objective function h describes a user-defined performance metrics, such as volumetric productivity, which can depend on the initial states (e.g. needed for determining product yields) and the final state vector at process end $(\widehat{t}_{\text{end}})$, the latter obtained from the simulated ODE model or its analytical solution. Additional constraints for the final state can be expressed by the last inequality.

For solving the optimization problem (9), the OptMSP package allows the user to choose between a brute-force approach (testing all possible realizations of MSPs with discretized switching times) or a directed optimization approach delivering the selected stages together with their order and switching times. Both search algorithms (brute-force or directed optimization) can handle arbitrary ODE models or explicit analytical models including the ones shown above and return a ranked list of MSPs (ranking based on the objective function) together with their associated characteristics, e.g., amount of biomass, substrate and product at the end of the process (time until substrate was consumed) or general performance parameters such as volumetric productivity.

Another metaparameter of OptMSP (beyond $s_{\max}$) is the minimal duration of a stage; this feature provides the user with more flexibility to incorporate preliminary knowledge (e.g., avoiding too short duration of stages that would be not realistic). The brute-force method requires in addition the specification of the discretization interval (t_{step}) of the switching time and the directed optimization the number of iterations.

2.2. Implementation details of the OptMSP package

The OptMSP package is a Python-based collection of functions for optimizing an MSP based on given ODE (or analytical) models for each stage according to the workflow in Fig. 1. OptMSP supports both brute-force testing as well as directed optimization to find an optimal MSP. OptMSP can combine arbitrary many stages, hence, 2-stage, 3-stage, 4-stage … processes are possible as result if they maximize the objective function.

Regarding the required model definition for each module, the user may easily adapt the default process models (either ODE-based (Eqn. 2) or as analytical solution (Eqs. (6-8)) or may provide his own set of models. In the latter case, he or she needs to implement a new function for each model that accepts the end and start time as well as the initial values of its state variables and returns the state variables at $\widehat{t}_{\text{end}}$. The set of state variables must include biomass, product or substrate, but optionally also further compounds or process parameters needed for a proper process description of the module (e.g. concentration of other byproducts).

The directed optimization strategy is based on an improved harmonic search (IHS) algorithm from the pygmo (Biscani and Izzo, 2020) package that was already successfully used in bioprocess design (Hemmerich et al., 2021). The pygmo package offers a logger functionality for the optimization algorithms that can be used to save the solution of each iteration for later analysis and ranking.

All calculations presented in the Results section have been performed via Python scripts and Jupyter notebooks under a custom Anaconda environment. The OptMSP package including a detailed documentation and tutorial Jupyter notebooks as well as all scripts, generated data and process models from the case studies in the Results section can be found in the OptMSP GitHub repository: https://github.com/klamt-lab/OptMSP.

3. Results

3.1. Example: MSP for lactate production with E. coli

In our first case study, which was motivated by the work of Wichmann et al. (2023), we considered the optimization of lactate production from glucose by *E. coli*, where the process modules represent different variants of aerobic growth phases (with low or no production of lactate) and anaerobic production phases (with or without growth). We used data from Wichmann et al. (2023), where different oxygen-dependent promoters were tested in *E. coli.* Generally, oxygen-dependent promoters are promising tools for MSPs since growth phases are often conducted in aerobic settings, while the production phase is usually carried out under anaerobic conditions. Oxygen-dependent promoters enable the induction of certain metabolic pathways or modules when turning off the oxygen supply during the switch to anaerobic conditions.

As background strain, Wichmann et al. (2023) considered an *E. coli* strain producing lactate as main fermentation product (ethanol and acetate production pathways deleted), in the following denoted by Eco$_{\text{lac}}$. Based on Eco$_{\text{lac}}$, another strain, Eco$_{\text{lac,+ATPase}}$, was constructed containing a plasmid harboring the genes of the F$_1$-ATPase, an enzyme hydrolyzing ATP to ADP. These genes were put under control of the oxygen-dependent promoter nirB-m, thus enabling induction of enforced ATP wasting (increasing ATP turnover) under anaerobic conditions. Enforced ATP wasting has been shown to increase specific production rates (but decreasing growth rates) (Boecker et al., 2019, 2021; Hädicke et al., 2015). For both strains, three different process modules are thus available with glucose as substrate: (1) aerobic growth (where no or very low lactate production takes place), (2) anaerobic growth and production, and (3) anaerobic production under growth arrest (nitrogen limitation). The specific rates μ, r_S and r_P were measured for each process (summarized in Table 1) and herein used to build the respective analytical (Eqs. (3)-(5)) and, for comparison, the ODE (Eqs. (6)-(8)) model for each module.

We then used OptMSP to find an optimal MSP maximizing the volumetric productivity. For demonstration purposes, we initially allowed the algorithm to use any of the six modules shown in Table 1, although this also admits impractical solutions where the resulting MSP can switch between the two strains (these MSPs are marked with an asterisk in Table 2). We applied both the brute-force method as well as directed optimization to determine an optimal MSP (t_{end}=24 h, $x_{B,0}$=0.01 gDW/L, $x_{S,0}$=100 mmol/L, $x_{P,0}$=0 mmol/L). The results are shown in Table 2. We first determined the best single-stage process, which is module 2 (growth-coupled production under anaerobic conditions with strain Eco$_{\text{lac}}$) resulting in a productivity of 9.55 mmol/L/h. When allowing a combination of two stages, the best MSP found by the brute-force method (with t_{step}=0.25 h) uses in the first phase module 1 (aerobic growth of the Eco$_{\text{lac}}$ strain) and switches after 4.5 h to module 5 (anaerobic stage with active ATPase) yielding a volumetric productivity of 20.013 mmol/L/h (see Table 2 and Fig. 3). Directed optimization found the same MSP but indicates that the optimal switching time point is at 4.61 h (leading to a slightly higher value for the maximal productivity), which cannot be found by the brute-force approach with the used discretization step (regarding optimal handling of the step size see also Discussion section). Generally, the optimal two-stage process is by far better than the found best OSP increasing the volumetric productivity by more than 100%.

When allowing combination of three stages, the optimal MSP indeed combines three modules (aerobic growth without ATPase, aerobic growth with active ATPase followed by growth-coupled production under anaerobic conditions) resulting in a slightly higher productivity compared to the best two-stage process (20.091 mmol/L/h (brute-force approach) or 20.092 mmol/L/h (directed optimization)).

As already mentioned above, the found optimal MSPs combine un-

Table 1
Process parameters of the different process modules (taken from Wichmann et al. (2023).

Process module	Strain	O$_2$	ATPase active	Growth	μ [h^{-1}]	r$_{\text{Glucose}}$ [mmol g$_{\text{DW}}^{-1}$ h^{-1}]	r$_{\text{Lactate}}$ [mmol g$_{\text{DW}}^{-1}$ h^{-1}]
1	Eco$_{\text{lac}}$	+	-	+	0.52	7.62	0.12
2		-	-	+	0.13	16.13	26.11
3		-	-	-	0.0	0.86	1.42
4	Eco$_{\text{lac,+ATPase}}$	+	-	+	0.50	6.08	0.08
5		-	+	+	0.06	22.86	41.25
6		-	+	-	0.0	9.37	17.61

Table 2
Results of different optimization scenarios for 1-stage, 2-stage, and 3-stage processes calculated by brute-force approach or directed optimization, each with the analytic and ODE-based model implementations.

Stages	Optimization method	Model	Module combination	t_{switch} [h]	q$_{\text{Lac}}$ [mmol/L/h]	Run time [min:sec]
1-stage	Brute force	Analytic	[2]	-	9.549	0:02
2*-stage	Brute force (t_{step}=0.25 h)	Analytic	[1, 5]	[4.50]	20.013	0:02
		Numeric (ODE)	[1, 5]	[4.50]	20.013	4:54
	Directed optimization (iterations= 10,000)	Analytic	[1, 5]	[4.61]	20.029	0:01
		Numeric (ODE)	[1, 5]	[4.61]	20.029	16:22
2-stage	Brute force (t_{step}=0.25 h)	Analytic	[4, 5]	[4.75]	19.905	0:02
		Numeric (ODE)	[4, 5]	[4.75]	19.905	2:25
	Directed optimization (iterations= 5000)	Analytic	[4, 5]	[4.86]	19.920	0:01
		Numeric (ODE)	[4, 5]	[4.86]	19.920	16:52
3*-stage	Brute force (t_{step}= 0.25 h)	Analytic	[1, 4, 5]	[3.25, 4.75]	20.091	1:04
		Numeric (ODE)	[1, 4, 5]	[3.25, 4.75]	20.091	926:30
	Directed optimization (iterations= 100,000)	Analytic	[1, 4, 5]	[3.28, 4.72]	20.092	0:03
		Numeric (ODE)	[1, 4, 5]	[3.28, 4.72]	20.092	121:42
3-stage	Brute force (t_{step}= 0.25 h)	Analytic	[4, 5]	[4.75]	19.905	0:19
		Numeric (ODE)	[4, 5]	[4.75]	19.905	247:64
	Directed optimization (iterations= 50,000)	Analytic	[4, 5]	[4.86]	19.920	0:04
		Numeric (ODE)	[4, 5]	[4.86]	19.920	134:17

Two different scenarios were considered for the optimization of MSPs with two or three stages. First, all six modules (1–6) could be combined (marked by 2*-stage and 3*-stage, respectively). In the second scenario, optimal 2-stage and 3-stage processes were separately determined for strain Eco$_{\text{lac}}$ (modules 1–3) and then for strain Eco$_{\text{lac,+ATPase}}$ (modules 4–6) and the run time of the two optimizations was added and the best combination of both strains selected.

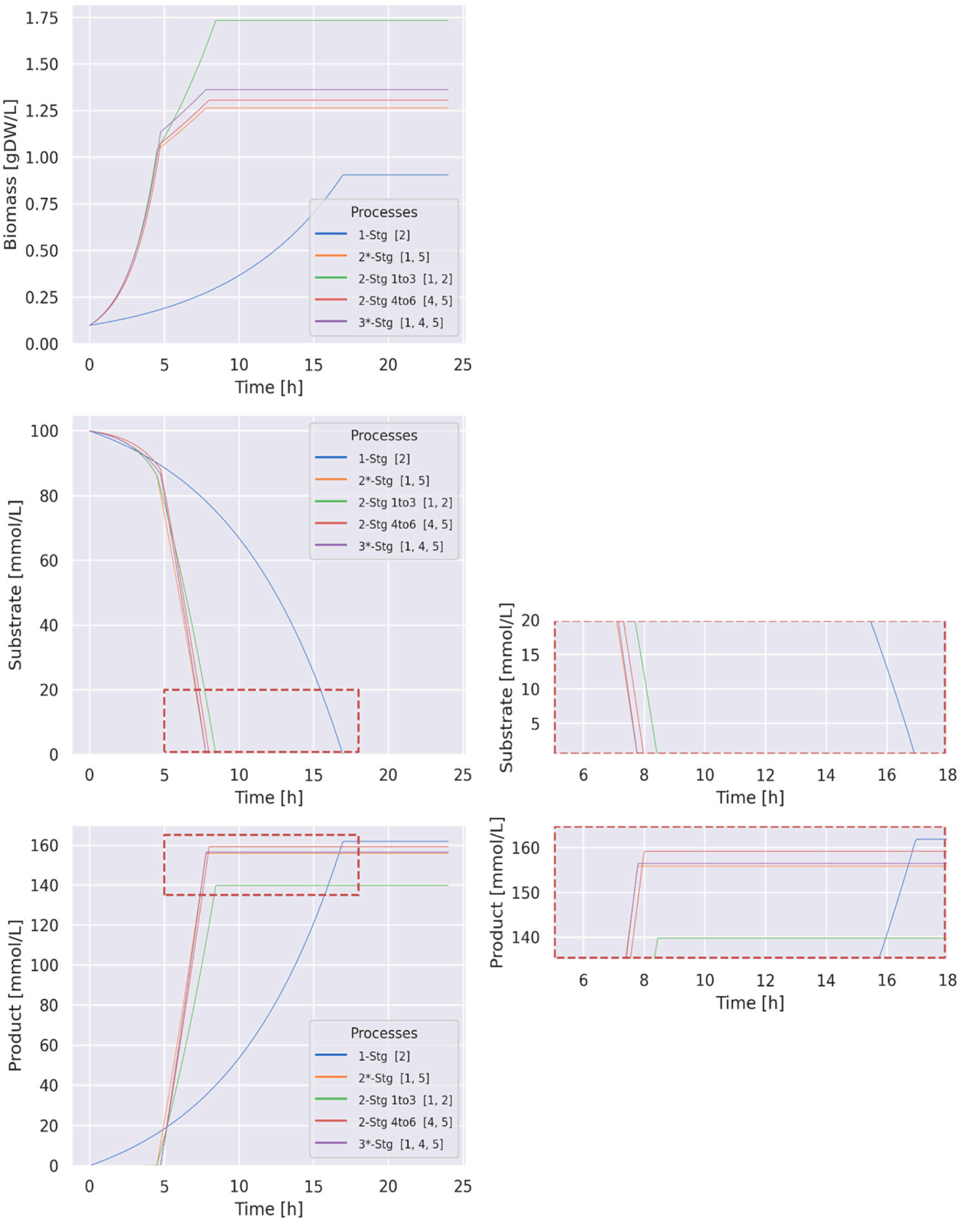

Fig. 3. Simulations of optimal MSPs as determined by the brute force approach (cf. Table 2).

realistically process modules from both strains. If we allow only combinations of modules 1–3 (strain Eco$_{lac}$) or of modules 4–6 (strain Eco$_{lac,+ATPase}$) then we find that the best two-stage process utilizes the ATPase strain with an aerobic growth phase (where expression of the ATPase genes in the ATPase strain is switched off by the oxygen-dependent promoter) followed by a growth-coupled anaerobic production phase (where the ATPase is active increasing the specific glucose uptake and lactate formation rates) with a t_{switch} at 4.75 h in the case of brute force and 4.86 h in the case of directed optimization (see 2-Stage (without asterisk) in Table 2). Allowing a three-stage process that can employ either the three modules 1–3 or the modules 4–6 does not improve the productivity; the optimization therefore delivers the same optimal two-stage process (compare 2-Stage and 3-stage (both without asterisk) in Table 2).

Table 2 also indicates the runtimes for all calculations. With the chosen discretization step for the brute-force approach, directed optimization is in most cases faster, although the number of iterations had to be increased up to 100,000 for three-stage calculations to find and converge to the MSP. While the runtimes remain low for both methods when using the analytical solution, using the ODE model, requiring numerical integration for each iteration, drastically increases the runtime by two to three (in one case even four) orders of magnitude. As expected, allowing combination of three instead of two stages increases the runtime significantly, especially for the brute-force approach since the number of module combinations (variations of three stages out of six modules) to be tested is massively higher. This effect is generally less pronounced when using the analytical solution, in fact, directed optimization is then only slightly slower when comparing two-stage and three-stage calculations.

3.2. *Example: optimizing plasmid DNA production in a fed-batch process with two phases*

The second example relates to a recent study by Gotsmy et al. (2023), which aimed to maximize the production of plasmid DNA (pDNA) through different process strategies. In their work it was shown that an increase of yield and volumetric productivity of pDNA production can be achieved by applying a three-stage fermentation process in which the first stage was a batch setup (that was given as fixed module and thus excluded from the optimization) and a fed-batch setup that was in turn separated into a growth and a production phase (the switching time between the two phases can be adjusted by the amount of sulfate added to the initial medium, which becomes limiting in the production phase). As in (Gotsmy et al., 2023), we here focused on optimal switching between the second and the third stage, i.e. between growth and production during fed-batch operation. In the following we will refer to these two phases as first and second phase (of the fed-batch part). We implemented the given ODE models of the two phases and used OptMSP to find the optimal switching time maximizing the volumetric productivity. With the obtained solution, we could reproduce the main results shown in Fig 4 (A and B) of Gotsmy et al. (2023). In particular, our calculations resulted in a maximal volumetric productivity of 0.206 g/L/h when the fed-batch switches at 21.00 h from growth to production, which corresponds to an initial sulfate concentration of 1.7 g/L (optimization with κ_{pDNA}=200%). This is in agreement with Fig. 4 (A) in (Gotsmy et al., 2023) where the optimal t^* (process duration of the second phase) is around 15 h. With an assumed process duration of 36 h for the two fed-batch phases, this corresponds to 21 h for the first (growth) phase, as was also determined by OptMSP.

A Jupyter notebook of this case study can be found at the OptMSP GitHub repository (https://github.com/klamt-lab/OptMSP).

3.3. *Example: optimal two-stage production with demanded minimal product yield*

As a third case study, we reproduced the results of Klamt et al. (2018). In this example, the goal was to find optimal switching times for maximizing the volumetric productivity of lactate production with *E. coli* in a two-stage process for different metabolic scenarios and, as a particular feature, under the side constraint of a demanded minimal product yield. The related Jupyter notebook can also be found in the OptMSP GitHub repository.

4. Discussion

In summary, in this study we introduced OptMSP, a new Python-based toolbox for finding optimal multi-stage bioprocesses in batch settings. Our package is based on a set of distinct and fully characterized sub-processes, whose dynamic behavior is described by process models, either via ODEs or via analytical solutions. This approach makes it distinct from methods like mcPECASO (Raj et al., 2020), which seek to predict the behavior of (e.g. engineered) cells based on constraint-based models to find optimal MSPs and associated strains. While the latter approach is useful to screen for potentially suitable strain designs, according to our experience it is generally difficult to predict basic growth characteristics, such as growth or substrate uptake rates, for engineered strains. In practice, one usually needs to decide for one or few strain designs (which may include integration of genetic parts to facilitate metabolic switches) and the constructed strains are then routinely characterized for their behavior under relevant growth conditions. Using these data to build associated process models enables a better predictivity of their behavior in single and (later) multi-stage processes. In the lactate case study, we used relatively simple process models, based on the measured specific rates of growth, substrate uptake and product formation for the different conditions. While this approach is a rough approximation of the process dynamics, it allows even usage of an analytical description of the dynamic process behaviors and could be useful and sufficient for many realistic applications. However, users retain the flexibility to introduce more intricate models if this is deemed necessary. This incorporation may also encompass additional functionalities, including, for example, transitional phases between sub-processes, thereby facilitating the representation of more realistic dynamics of the process. For example, as a simplifying assumption, in our first case study we did not consider a transition phase when switching from aerobic to anaerobic conditions since the data of Wichmann et al. (2023) indicated a relatively short transition time (20–30 min) in which the cells switched their metabolism from aerobic to anaerobic operation.

OptMSP offers two distinct methods to identify the optimal MSP: the brute-force and the directed optimization approach. The brute-force method permits exhaustive testing of all conceivable combinations of modules along discretized switching times. The advantage of such an exhaustive approach is that the entire solution space is scanned resulting in a ranked list of all possible MSPs (within the discretized time steps). This full list of MSPs can then also be re-ranked if another objective function is considered or if complex constraints for the allowable solutions are involved. Potential problems of the brute-force approach are related to the discretization step size t_{step}. The best choice of t_{step} depends on the complexity of the specific process and its modules (including number of modules, complexity and size of the models (correlating with time needed for simulation), and process duration). Generally, one will try to make t_{step} as small as possible to obtain a fine resolution of the sampled solution space and to thus get close to the optimal solution. However, if t_{step} is too small the calculations will take very long. A useful strategy is to start the optimization with a larger t_{step} and then to decrease it successively as long as the runtime is acceptable. In a final run, an optimization could be performed with an even smaller t_{step} at the vicinity of the switching times of the optimal solution found in the first rounds of optimization.

In contrast, the directed optimization approach facilitates the consideration of switching times as continuous parameters thus avoiding the potential problems of time discretization. However, even global (directed) optimization solvers cannot guarantee to identify the optimal MSP. When testing MSPs incorporating more than four stages, the prudent choice is to opt for the directed optimization approach due to combinatorial explosion massively increasing run time. However, consideration of more than four stages are not relevant for most applications. Generally, if complex models or higher order MSPs are considered, using a parallelization framework could become an option when using the brute-force approach.

Other reported methods for designing multi-stage or more general dynamically controlled bioprocesses are based on dynamic optimization approaches (Banga et al., 2005; Gadkar et al., 2005; Anesiadis et al., 2013; Hjersted and Henson, 2006; Jabarivelisdeh and Waldherr, 2018; Ryu et al., 2019; Espinel-Ríos et al., 2022). In these formulations, one is typically given an integrated process model in which certain static (e.g. initial conditions) as well as dynamic variables (such as temperature or oxygen supply) can be adjusted to obtain a process with optimal behavior and performance. The designed process involves then usually continuous changes in the external inputs instead of distinct, well-defined stages as considered by OptMSP. However, this requires often more complex process models (e.g. to describe how dynamic changes of the input variables translate to changes in the metabolic fluxes (Gadkar et al., 2005; Anesiadis et al., 2013; Jabarivelisdeh and Waldherr, 2018; Espinel-Ríos et al., 2022) and solving such dynamic (and partially bilevel) optimization problems is much more complicated requiring sophisticated solvers and expert knowledge in adjusting associated metaparameters. We believe that our presented OptMSP approach and associated software provides a good compromise of flexibility and simplicity which makes it a useful tool that can straightforwardly be employed for many relevant applications where distinct phases of processes can be combined. On the other hand, although discrete changes in feeding could be taken into account when switching between modules in the OptMSP approach, if one aims for the optimization of fed-batch processes dynamic optimization approaches (Banga et al., 2005; Hjersted and Henson, 2006) will become the method of choice since they allow proper consideration of dynamic feeding profiles (this was not relevant in our second case study, where a constant feeding was assumed during fed-batch operation).

In the larger first case study, we employed and tested our OptMSP toolbox to identify the MSP with maximal volumetric productivity for lactate production, where the six available modules represented two dedicated *E. coli* strains (with or without inducible operation of the ATPase to enforce higher ATP turnover), each cultivatable in three different stages (aerobic growth, anaerobic growth-coupled or growth-decoupled production of lactate). In our investigation we benchmarked the brute-force as well as the directed optimization method, each utilizing both the analytical and the ODE models, hence, four different computational procedures were tested for each MSP optimization scenario. For all scenarios tested, we found that the different calculation schemes found consistent results, while the runtimes differed partially significantly. Generally, using the analytical solution of the simple process model largely improves runtime behavior of both brute-force testing and directed optimization, while the latter performed superior in the majority of cases. Interestingly, the analysis of a three-stage MSP showed that the best process indeed uses three different modules to reach optimal behavior demonstrating that optimal MSPs may involve more than two stages. However, this hypothetical result allowed a switch between both strains, a scenario which is infeasible in practice. When module combinations where only allowed for modules from a single strain, then we determined that the most favorable process is a two-stage configuration of the ATPase strain, encompassing an aerobic growth phase (where the ATPase is switched off in that strain) succeeded by an anaerobic phase with growth-coupled production where the ATPase and thus ATP wasting is active. Interestingly, this result is in line with the (*in silico*) prediction made in (Raj et al., 2020), that a two-stage process with growth-coupled production in the second stage is most likely superior to a pure production process without growth. Indeed, while module 6 (production under growth arrest) leads to higher product yields (1.88 mol lactate / mol glucose; Table 1) compared to growth-coupled production in module 5 (1.80 mol lactate / mol glucose), the specific productivity of sub-process 5 is ca. 130% higher than under module 6 (due to increased metabolic activity in growing cells), which is favorable for achieving the highest volumetric productivity.

In the other two case studies we exemplified the use of OptMSP for different setups (e.g. fed-batch) and for additional constraints (e.g. demanded minimal product yield) further demonstrating the various options of OptMSP to design optimal multi-stage processes.

CRediT authorship contribution statement

Jasmin Bauer: Methodology, Implementation, Investigation, Validation, Writing – original draft, Writing – review & editing, Visualization. **Steffen Klamt:** Conceptualization, Methodology, Supervision, Funding acquisition, Writing – original draft, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

All relevant data that support the findings are available within this article. The computer code, models and tutorials used herein together with a documentation for OptMSP can be found in the OptMSP GitHub repository: https://github.com/klamt-lab/OptMSP.

Acknowledgements

This work was supported by the International Max Planck Research School for Advanced Methods in Process and Systems Engineering (IMPRS ProEng) and by the state of Saxony-Anhalt within the SmartProSys initiative. We thank Mathias Gotsmy and Jürgen Zanghellini for sharing their ODE models for their investigated fed-batch process for plasmid DNA production. Additionally, we thank Pavlos Stephanos Bekiaris for the code evaluation, tests for reproducibility of results and helpful comments on the project.

References

Anesiadis, N., Kobayashi, H., Cluett, W.R., Mahadevan, R., 2013. Analysis and design of a genetic circuit for dynamic metabolic engineering. ACS Synth. Biol. 2 (8), 442–452.

Banga, J.R., Balsa-Canto, E., Moles, C.G., Alonso, A.A., 2005. Dynamic optimization of bioprocesses: Efficient and robust numerical strategies. J. Biotechnol. 117 (4), 407–419.

Biscani, F., Izzo, D., 2020. A parallel global multiobjective framework for optimization: pagmo. J. Open Source Softw. 5 (53), 2338.

Boecker, S., Zahoor, A., Schramm, T., Link, H., Klamt, S., 2019. Broadening the scope of enforced ATP wasting as a tool for metabolic engineering in Escherichia coli. Biotechnol. J. 14 (9), 1800438.

Boecker, S., Harder, B.-J., Kutscha, R., Pflügl, S., Klamt, S., 2021. Increasing ATP turnover boosts productivity of 2, 3-butanediol synthesis in Escherichia coli. Microb. Cell Factor. 20 (1), 12.

Burg, J.M., Cooper, C.B., Ye, Z., Reed, B.R., Moreb, E.A., Lynch, M.D., 2016. Large-scale bioprocess competitiveness: the potential of dynamic metabolic control in two-stage fermentations. Curr. Opin. Chem. Eng. 14, 121–136.

Carrasco-López, C., García-Echauri, S.A., Kichuk, T., Avalos, J.L., 2020. Optogenetics and biosensors set the stage for metabolic cybergenetics. Curr. Opin. Biotechnol. 65, 296–309.

Espinel-Ríos, S., Morabito, B., Pohlodek, J., Bettenbrock, K., Klamt, S., Findeisen, R., 2022. Optimal control and dynamic modulation of the ATPase gene expression for enforced ATP wasting in batch fermentations. IFAC-Pap. 55 (7), 174–180.

Farmer, W.R., Liao, J.C., 2000. Improving lycopene production in Escherichia coli by engineering metabolic control. Nat. Biotechnol. 18 (5), 533–537.

Gadkar, K.G., Doyle Iii, F.J., Edwards, J.S., Mahadevan, R., 2005. Estimating optimal profiles of genetic alterations using constraint-based models. Biotechnol. Bioeng. 89 (2), 243–251.

Gotsmy, M., Strobl, F., Weiß, F., Gruber, P., Kraus, B., Mairhofer, J., Zanghellini, J., 2023. Sulfate limitation increases specific plasmid DNA yield and productivity in E. coli fed-batch processes. Microb. Cell Factor. 22 (1), 242.

Gupta, A., Reizman, I.M.B., Reisch, C.R., Prather, K.L., 2017. Dynamic regulation of metabolic flux in engineered bacteria using a pathway-independent quorum-sensing circuit. Nat. Biotechnol. 35 (3), 273–279.

Hädicke, O., Bettenbrock, K., Klamt, S., 2015. Enforced ATP futile cycling increases specific productivity and yield of anaerobic lactate production in Escherichia coli. Biotechnol. Bioeng. 112 (10), 2195–2199.

Harder, B.J., Bettenbrock, K., Klamt, S., 2018. Temperature-dependent dynamic control of the TCA cycle increases volumetric productivity of itaconic acid production by Escherichia coli. Biotechnol. Bioeng. 115 (1), 156–164.

Hemmerich, J., Tenhaef, N., Wiechert, W., Noack, S., 2021. pyFOOMB: Python framework for object oriented modeling of bioprocesses. Eng. life Sci. 21 (3-4), 242–257.

Hjersted, J.L., Henson, M.A., 2006. Optimization of fed-batch Saccharomyces cerevisiae fermentation using dynamic flux balance models. Biotechnol. Prog. 22 (5), 1239–1248.

Hwang, H.J., Kim, J.W., Ju, S.Y., Park, J.H., Lee, P.C., 2017. Application of an oxygen-inducible nar promoter system in metabolic engineering for production of biochemicals in Escherichia coli. Biotechnol. Bioeng. 114 (2), 468–473.

Jabarivelisdeh, B., Waldherr, S., 2018. Optimization of bioprocess productivity based on metabolic-genetic network models with bilevel dynamic programming. Biotechnol. Bioeng. 115 (7), 1829–1841.

Klamt, S., Mahadevan, R., Hädicke, O., 2018. When do two-stage processes outperform one-stage processes? Biotechnol. J. 13 (2), 1700539.

Menacho-Melgar, R., Ye, Z., Moreb, E.A., Yang, T., Efromson, J.P., Decker, J.S., et al., 2020. Scalable, two-stage, autoinduction of recombinant protein expression in E. coli utilizing phosphate depletion. Biotechnol. Bioeng. 117 (9), 2715–2727.

Raj, K., Venayak, N., Mahadevan, R., 2020. Novel two-stage processes for optimal chemical production in microbes. Metab. Eng. 62, 186–197.

Ryu, K.H., Kim, B., Lee, J.H., 2019. A model-based optimization of microalgal cultivation strategies for lipid production under photoautotrophic condition. Comput. Chem. Eng. 121, 57–66.

Soma, Y., Hanai, T., 2015. Self-induced metabolic state switching by a tunable cell density sensor for microbial isopropanol production. Metab. Eng. 30, 7–15.

VanRossum G., Drake F.L. The python language reference: Python Software Foundation Amsterdam, Netherlands; 2010.

Venayak, N., Von Kamp, A., Klamt, S., Mahadevan, R., 2018. MoVE identifies metabolic valves to switch between phenotypic states. Nat. Commun. 9 (1), 5332.

Wichmann, J., Behrendt, G., Boecker, S., Klamt, S., 2023. Characterizing and utilizing oxygen-dependent promoters for efficient dynamic metabolic engineering. Metab. Eng. 77, 199–207.